自然语言处理实践

聊天机器人技术原理与应用

王昊奋　邵浩　李方圆　张凯　宋亚楠　编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

VISIT…

内 容 简 介

聊天机器人作为人工智能技术的杀手级应用，发展得如火如荼，各种智能硬件层出不穷。本书系统地阐述了聊天机器人的分类和关键技术，不仅给出了实际案例，还展望了聊天机器人在通往更智能化、更人性化、更趣味化的道路上所面临的挑战。同时，针对聊天机器人在从感知智能到认知智能的跨越中所面临的难题，本书着重讨论了知识图谱和深度学习技术在自然语言处理、问答、推理、服务融合等方面的应用。

本书适合有志于从事人工智能行业，以及想了解聊天机器人到底是什么的读者阅读。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

自然语言处理实践：聊天机器人技术原理与应用 / 王昊奋等编著. —北京：电子工业出版社，2019.3
ISBN 978-7-121-35715-2

Ⅰ. ①自… Ⅱ. ①王… Ⅲ. ①自然语言处理－研究②智能机器人－研究 Ⅳ. ①TP391②TP242.6

中国版本图书馆 CIP 数据核字(2018)第 266347 号

策划编辑：郑柳洁
责任编辑：郑柳洁
印　　刷：
装　　订：
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编：100036
开　　本：720×1000　1/16　印张：12.75　字数：191 千字
版　　次：2019 年 3 月第 1 版
印　　次：2019 年 3 月第 1 次印刷
定　　价：69.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

推荐序

聊天机器人是社会关系网络、自动客服、语音助手、智能音箱、游戏等的重要支撑技术，它综合应用了自然语言处理技术。自然语言处理是体现语言智能非常关键的技术，它分析、理解和生成自然语言，实现人与机器的自然交流。同时，机器翻译、自动文摘、自动写作、邮件或者短消息的自动回复也有助于人与人之间的交流。如果语言智能可以实现突破，跟它同属认知智能的知识图谱与常识推理等技术也会得到长足的发展，并推动整个人工智能体系的进步，使更多的场景落地。自然语言处理被视为人工智能“皇冠上的明珠”。要做好这项技术，达到和人一样自然的交互是非常具有挑战性的一项课题。许多积极投身于自然语言处理研究和开发的同仁，迫切需要掌握自然语言处理的基础技术，了解技术前沿。

我很高兴看到本书的出版。它系统地介绍了聊天机器人的技术体系和自然语言处理在聊天机器人中的应用，辅以案例，理论和实践结合良好。本书深入浅出的风格对不同层次的读者都有帮助。

本书由王昊奋博士和邵浩博士主导，他们二位都是从学术界跨越到工业界的年轻人，并致力于将技术应用到产品实践中。我和王昊奋在中国计算机学会

术语工作委员会和自然语言专委会等组织中有紧密的合作。我认为，他不仅在学术上积极进取，还特别希望尝试把各种新技术应用到产品中。他将理论和实践相结合，多年来积累了丰富的研发经验，走出了一条独到的创新之路。本书由多位相关企业的资深技术研发人员参与撰写。因此我相信本书一定会激发大家对聊天机器人的兴趣和更深入的思考。

从本书的内容上看，除了对聊天机器人的历史发展和技术体系的阐述，重点介绍了聊天机器人的 3 种典型表现形式：闲聊、对话和问答。以闲聊型聊天机器人为例，虽然基于检索的方法是目前主流的产品实现方式，但随着自然语言处理端到端技术的发展，生成式对话越来越受重视，有很多研究者尝试用生成方法解决个性化、多轮对话和安全回复等问题。同时，本书介绍了知识图谱的重要作用,因为基于知识图谱的问答也是问答型聊天机器人的重要组成部分。本书尽可能完整地展现了聊天机器人相关技术的最新进展，有兴趣的读者可通过此书全面了解聊天机器人。

聊天机器人已经在智能客服、知识问答等场景里有了较好的应用，未来会在大数据、深度学习和重要场景的推动下进一步提升智能水平。我们可以畅想，在未来的某个时刻，会出现一个基于人工智能技术的虚拟生命，它能够真正理解人类的语言，有自己的记忆和情感，并可以和人进行自然真实的对话。尽管我们离这个目标尚远，但是我们可以逐步靠近。这里孕育着无穷的研究、开发机会和乐趣。我期待本书能激励更多优秀的年轻人投身其中，做出更多成就！

微软亚洲研究院副院长、国际计算语言学会主席

周明

前　言

缘起

写作本书的初衷，是作者在聊天机器人相关的技术公司工作时，在做产品的过程中，深深感受到理论和实践的差别。举例来说，学术界一直追捧的 seq2seq 技术，并没有很好地应用在聊天机器人的产品中。而且，对于刚入职的工程师，也没有一本系统性的书籍来帮助他们快速理解和掌握聊天机器人的技术脉络。因此，我提议：我们为什么不写一本全面的技术读本，将我们在实践中遇到的问题和解决方案都放到这本书里，让更多的读者了解聊天机器人的知识体系，避免踩我们在工作中踩过的很多坑呢？于是我们开始了本书的撰写。希望这本书能尽可能全面地梳理聊天机器人技术，使读者更深入地理解其背后的理论知识。

本书特色

本书应该是国内第一本聊天机器人参考书，书中不仅介绍了聊天机器人的发展历史，还深入介绍了不同类型聊天机器人的技术实现。无论是拥有实体的聊天机器人还是聊天机器人软件，其功能都跳不出闲聊、问答、对话和主动交互 4 种。不同类型的聊天机器人的侧重点不同，但终极目标都是拥有自我感知能力，并能像人一样进行情感交互。本书涵盖的范围比较广泛，但受限于时间和精力，对某些特定的技术点，我们仅仅给出了简要介绍（例如语音识别和语音合成技术），而将主要精力放在了在文字层面聊天机器人如何进行交互上。

对于工业界的朋友，希望本书能够在您寻找特定技术的时候提供一定帮助；对于学术界的专家，本书给出的很多难题也期待着您在理论上加以研究并寻求突破。

本书分 7 章。第 1 章简要介绍了聊天机器人的发展和分类，第 2 章给出了聊天机器人的技术体系介绍，第 3 章到第 5 章分别介绍了 3 种不同类型的聊天机器人的技术实现（问答、对话和闲聊），第 6 章给出了聊天机器人系统评测的相关信息，第 7 章提出了聊天机器人进一步发展所面临的技术挑战和展望。下面简要介绍每章的具体内容：

第 1 章“聊天机器人概述” 在本章中，我们追溯了聊天机器人的发展历史，并阐述了聊天机器人的分类和应用场景，从技术层面给出了一个典型的聊天机器人应该包含的技术框架，同时着重介绍了最具代表性的聊天机器人产品。

第 2 章“聊天机器人技术原理” 在本章中，我们从技术的角度，详细介绍了一个文字型交互聊天机器人涉及的技术，包括自然语言理解、对话管理和自然语言生成。我们不仅介绍了传统的自然语言处理技术，也给出了深度学习在解决同类问题上的研究进展。同时，引出了跨越认知智能的关键技术之一——知识图谱，通过不同的例子，阐述了从构建到应用知识图谱的过程。

第 3 章“问答系统” 在本章中，我们介绍了聊天机器人的一种形式——问答。对于某一问题，问答系统旨在获取其精准答案。我们重点介绍了基于知识库的问答系统，阐述了构建知识库所需的技术，并给出了 IBM Watson 问答系统的详细说明。同时，我们介绍了 4 种主流的问答方法，包括模板匹配、语义解析、图遍历和深度学习。最后，给出了一个问答系统的具体实践案例。

第 4 章“对话系统” 在本章中，我们主要介绍面向任务的对话系统。与问答系统不同，面向任务的对话系统旨在完成用户指定的一项特定任务。从技术的层面，我们分别介绍了自然语言理解、对话状态跟踪、对话策略学习及自然语言生成，同时穿插了具体的案例，让读者可以有更直观的理解。

第 5 章“闲聊系统” 在本章中，我们分别介绍了闲聊系统的两种实现方

式，一种基于对话库检索，另一种基于生成模型。我们不仅介绍了技术的最新进展，还给出了具体的实现案例。

第 6 章“聊天机器人系统评测” 在本章中，我们梳理了目前国内外聊天机器人评测的公开会议、数据集和进展，并分别针对问答系统和对话系统给出了详细的评测介绍。

第 7 章“聊天机器人挑战与展望” 在本章中，我们给出了聊天机器人发展到现阶段所面临的挑战，并对未来不同场景的应用进行了展望。同时，对聊天机器人发展的下一代范式——虚拟生命，给出了我们的见解和期望。

本书是集体智慧的结晶，写作成员包括王昊奋、邵浩、李方圆、张凯、宋亚楠。同时，感谢很多同事和朋友在写作过程中给予的协助。在写作过程中，我们从实际出发，考虑搭建一个聊天机器人所需要的技术应该是什么样的，同时，关注国内外关于聊天机器人、自然语言处理、知识图谱、机器学习的最新进展，并思考如何将这些技术真正应用于构建聊天机器人中。需要说明的是，在写作过程中，我们参阅了很多领域专家的资料，并尽可能地将所有参考资料都列出了。如果您发现某些内容有争议，请联系我们。

尤其要感谢郑柳洁编辑，没有她的督促和协助，本书不可能有这样的完成度。

拥抱人工智能时代

最近几年，技术的飞速发展让我们每个人都无比兴奋。我们也很激动地看到 AI 巨头不断地开源最新、最快的模型，例如谷歌开源了语言模型 BERT，已经在所有 benchmark 数据集上取得了突破。“工欲善其事，必先利其器”，这些强大的算法和工具，让人工智能领域从业者可以创造出更多、更好的产品。在人工智能的发展过程中，我们希望贡献自己的微薄之力。如果读者能够在阅读的过程中获得一点灵感，也将让我们无比欣慰。

愿意创造下一代聊天机器人范式的朋友，我们非常诚挚地邀请您们，一起创造出让人惊艳的、跨越感知智能和认知智能的产品！

目 录

1 聊天机器人概述.....1

1.1 聊天机器人的发展历史.....1

1.2 聊天机器人的分类与应用场景.....6

1.3 聊天机器人生态介绍.....9

1.3.1 典型聊天机器人框架介绍.....11

1.3.2 聊天机器人平台介绍.....13

1.3.3 典型的聊天机器人产品介绍.....13

1.4 参考文献.....19

2 聊天机器人技术原理.....20

2.1 自然语言理解.....21

2.1.1 自然语言理解概览.....23

2.1.2 自然语言理解基本技术.....26

2.1.3 自然语言表示和基于深度学习的自然语言理解.....36

2.1.4 基于知识图谱的自然语言理解.....46

2.2 自然语言生成.....56

2.2.1 自然语言生成综述.....56

2.2.2 基于检索的自然语言生成.....58

2.2.3 基于模板的自然语言生成.....59

2.2.4 基于深度学习的自然语言生成 60
2.3 对话管理 61
2.4 参考文献 65
3 问答系统 67
3.1 问答系统概述 67
3.2 KBQA 系统 71
3.2.1 KBQA 系统简介 71
3.2.2 主流的 KBQA 方法 79
3.3 KBQA 系统实现 96
3.3.1 系统简介 96
3.3.2 模块设计 97
3.4 参考文献 105
4 对话系统 109
4.1 对话系统概述 109
4.2 对话系统技术原理 113
4.2.1 NLU 模块 115
4.2.2 DST 模块 120
4.2.3 DPL 模块 121
4.2.4 NLG 模块 126
4.3 基于聊天机器人平台搭建对话系统 126
4.3.1 NLU 模块实现 129
4.3.2 DST 与 DPL 模块实现 130
4.3.3 NLG 模块实现 131
4.4 面向任务的对话系统实现 132
4.5 参考文献 137
5 闲聊系统 139
5.1 闲聊系统概述 139
5.2 基于对话库检索的闲聊系统 140
5.2.1 基于对话库检索的闲聊系统介绍 140

5.2.2 对话库的建立 143
5.2.3 基于检索的闲聊系统实现 145
5.3 基于生成的闲聊系统 150
5.3.1 基于生成的闲聊系统介绍 150
5.3.2 生成式闲聊系统的新发展 152
5.3.3 基于生成的闲聊系统实现 155
5.4 参考文献 157
6 聊天机器人系统评测 159
6.1 问答系统评测 159
6.1.1 问答系统评测会议 160
6.1.2 问答系统评测数据集 171
6.1.3 问答系统评测标准 173
6.2 对话系统评测 174
6.2.1 对话系统评测会议 176
6.2.2 对话系统评测数据集 177
6.2.3 对话系统评测标准 178
6.3 闲聊系统评测 179
6.3.1 闲聊系统评测介绍 179
6.3.2 闲聊系统评测标准 180
6.4 参考文献 183
7 聊天机器人挑战与展望 185
7.1 开放式挑战 185
7.2 技术与应用展望 187
7.3 从聊天机器人到虚拟生命 190
7.4 参考文献 193

1 聊天机器人概述

1.1 聊天机器人的发展历史

聊天机器人，是一种通过自然语言模拟人类，进而与人进行对话的程序。它既可以在特定的软件平台（如 PC 平台或者移动终端设备）上运行，也可以在类人的硬件机械体上运行。聊天机器人已经有近 70 年的发展历史，让我们一同回顾半个多世纪以来，不同历史阶段典型的聊天机器人项目和产品。

1．聊天机器人溯源及发展（1950—1990 年）

对聊天机器人的研究可以追溯到 1950 年图灵（Alan M. Turing）在 *Mind* 期刊上发表的文章 *Computing Machinery and Intelligence*，这篇文章开篇就提出了“机器能思考吗？（Can machines think?）”的设问，然后提出通过让机器参与模仿游戏（Imitation Game）来验证“机器”能否进行“思考”，进而提出了经典的图灵测试（Turing Test）。通过图灵测试被认为是人工智能研究的终极

目标[①]，图灵本人也因而被称为“人工智能之父”。

已知的发布最早的聊天机器人程序 ELIZA[1]诞生于 1966 年，由麻省理工学院（MIT）的约瑟夫·魏泽鲍姆（Joseph Weizenbaum）开发。魏泽鲍姆是自然语言处理方面的先驱，他开发的 ELIZA 被看作可以用于临床模拟罗杰斯心理治疗的 BASIC 脚本程序。值得注意的是，尽管 ELIZA 的实现技术仅为对用户输入计算机的话语做关键词匹配，并且其回复规则是由人工编写的（魏泽鲍姆的本意只是让 ELIZA 模仿人类的交谈），但用户与 ELIZA 交谈时却如同自己面对着心理治疗师，开始向 ELIZA 倾诉自己内心深处的想法。随后，魏泽鲍姆撰写了 *Computer Power and Human Reason* 一书，以表达他对人工智能技术的看法。不论怎样，ELIZA 对自然语言处理和人工智能的研究与发展产生了重大影响，全球各地的研究机构也由此开始了对聊天机器人的相关研究。

1972 年，美国精神病学家肯尼思·科尔比（Kenneth Colby）在斯坦福大学（Stanford University）使用 LISP 编写了模拟偏执型精神分裂症表现的计算机程序 PARRY。由于 PARRY 体现的会话策略比魏泽鲍姆的 ELIZA 更严谨更先进，PARRY 被描述为“有态度的 ELIZA”。研究人员在 20 世纪 70 年代早期，使用图灵测试的变体对 PARRY 进行了测试，测试由一组经验丰富的精神科医生参与，这些参与测试的精神科医生通过电传打印机分别与患者和运行 PARRY 的计算机进行对话，并将这些对话记录展示给另一组（33 名）精神科医生。这两组精神科医生分别被要求确定哪些对话是人类患者产生的，哪些是计算机程序产生的。测试结果表明，参与测试的两组精神科医生中只有 48%在规定时间内做出了正确的判断，正确率约等于随机投票产生的正确率。

1988 年，英国程序员罗洛·卡彭特（Rollo Carpenter）创建了聊天机器人

① 虽然俄罗斯人弗拉基米尔·维西罗夫（Vladimir Veselov）创立的人工智能软件尤金·古斯特曼（Eugene Goostman）在 2014 年就通过了图灵测试，但聊天机器人离真正的“智能”还有很长的路要走。

Jabberwacky。Jabberwacky 项目的目标是“以有趣、娱乐和幽默的方式模拟自然的人际聊天”，这个项目也是通过与人类互动创造人工智能聊天机器人的早期尝试，但 Jabberwacky 并未被用于执行任何其他功能。Jabberwacky 项目于 1997 年正式上线，上线后它会存储所有用户与自己的对话，并且在与用户的对话过程中使用上下文模式匹配技术找到最合适的回复内容。Jabberwacky 并没有硬编码的规则，它完全依赖于反馈的原则，这一点与大多数基于规则约束的聊天机器人非常不同。

也是在 1988 年，加州大学伯克利分校（UC Berkeley）的罗伯特·威林斯基（Robert Wilensky）等人开发了名为 UC（UNIX Consultant）的聊天机器人系统。顾名思义，UC 聊天机器人的目的是帮助用户学习使用 UNIX 操作系统。UC 聊天机器人以英文回复用户，它具备分析用户输入、确定用户需求、给出解决用户需求的规划、决定与用户沟通的内容、根据用户对 UNIX 系统的熟悉程度进行建模等功能。如果说 ELIZA 开启了智能聊天机器人的研究，那么 UC 则真正提高了聊天机器人的智能化程度。

1990 年，美国科学家兼慈善家休·勒布纳（Hugh G. Loebner）设立了人工智能年度比赛——勒布纳奖（Loebner Prize）。勒布纳奖旨在借助交谈测试机器的思考能力，它被看作对图灵测试的一种实践，其比赛的奖项分为金、银、铜三等。根据勒布纳奖的规定，如果参与比赛的程序不仅能通过以文本方式进行的交谈测试，还能在音频和视频测试中过关，则获金奖，赢得 10 万美元和一枚 18K 黄金制金牌，同时勒布纳奖的年度比赛将会中止；如果程序能在以文本方式进行的交谈测试中长时间迷惑住至少半数裁判，则获银奖；如果程序未达到以上标准，则在测试中迷惑住最多裁判的程序赢得 2000 美元和一枚铜奖。从 1991 年首届比赛开办，至本书撰写时，尚无参赛程序达到金奖或者银奖标准。

2．聊天机器人研究兴起（1990—2010 年）

在勒布纳奖的推动下，聊天机器人迎来了研究的高潮，其中较有代表性的聊天机器人系统是 1995 年 12 月 23 日诞生的 ALICE[①]（Artificial Linguistic Internet Computer Entity）。受到相关聊天机器人研究的启发，人工智能科学家理查德·华勒斯（Richard S. Wallace）博士在 1995 年开发了 ALICE 系统，ALICE 可以通过和网民的聊天不断学习，同时基于大量的预置问答模板回答用户的问题。虽然 ALICE 曾在 2000 年、2001 年和 2004 年三次获得勒布纳奖，但其和 ELIZA 并无本质区别（ALICE 仍然完全基于模板匹配技术，不具备逻辑推理能力）。理查德·华勒斯博士将 ALICE 安装到网络服务器，并于 1998 年开源了该项目，目前全世界已有超过 500 个开发者为 ALICE 项目贡献代码。尽管 ALICE 采用的是启发式模板匹配的对话策略，但是它仍然被认为是同类型聊天机器人中性能最好的系统之一。图 1-1 所示为笔者与 ALICE 聊天的截图。值得一提的是，随 ALICE 一同发布的 AIML（Artificial Intelligence Markup Language）目前在移动端虚拟助手的开发中得到了广泛的应用。

图 1-1　笔者与 ALICE 的聊天记录

① http://www.alicebot.org/

2001 年，SmarterChild 在短信和即时聊天工具中的广泛流行，使得聊天机器人第一次被应用在了即时通信领域。2006 年，IBM 开始研发能够用自然语言回答问题的最强大脑 Watson，作为一台基于 IBM “深度问答” 技术的超级计算机，Watson 能够采用上百种算法在 3 秒内找出特定问题的答案。

3．聊天机器人研究方兴未艾（2010 年至今）

2010 年，苹果公司推出了人工智能助手 Siri，Siri 的技术来源于美国国防部高级研究规划局公布的 CALO 计划：一个简化军方繁复事务，且具备学习、组织及认知能力的虚拟助理。CALO 计划衍生出来的民用版软件就是 Siri 虚拟个人助理。

此后，微软小冰、微软 Cortana（小娜）、阿里小蜜、京东 JIMI、网易七鱼等各类聊天机器人层出不穷，并且这些聊天机器人逐渐渗透进人们生活的各个领域。

2016 年，全球各大公司开始推出可用于聊天机器人系统搭建的开放平台或开源架构。

2010 年至今，标志性的聊天机器人产品如图 1-2 所示。

图 1-2　标志性的聊天机器人产品

聊天机器人的发展历史说明，人类从未放弃将聊天机器人作为人机交互工具的研究。特别是近几年，随着人工智能相关技术“东风”渐起，自然语言处理研究硕果颇丰，聊天机器人相关技术迅速发展。同时，聊天机器人作为一种新颖的人机交互方式，正在成为移动搜索和服务的入口之一，毕竟搜索引擎的最终形态很可能就是聊天机器人。众多人工智能领域的探索者和开发者都想紧紧抓住并抢占聊天机器人这一新的交互入口。

1.2 聊天机器人的分类与应用场景

近年来，基于聊天机器人系统的应用层出不穷，下面我们从几个维度对其进行分类。

1．基于应用场景的聊天机器人分类

从应用场景的角度看，可以将聊天机器人分为在线客服、娱乐、教育、个人助理和智能问答 5 类。

在线客服聊天机器人系统的主要功能是自动回复用户提出的与产品或服务相关的问题，以降低企业客服运营成本、提升用户体验。其服务通常是以网站和手机终端为载体而实现的。代表性的商用在线客服聊天机器人系统有小 i 机器人、京东 JIMI 客服机器人、阿里小蜜等。以京东 JIMI 客服机器人为例，用户可以通过与 JIMI 聊天了解商品的具体信息、了解平台的活动信息、反馈购物中存在的问题等。另外，JIMI 具备一定的拒识能力，因此可以知道用户的哪些问题是自己无法回答的，且可以及时将用户转向人工客服。阿里巴巴集团在 2015 年 7 月 24 日发布了一款人工智能购物助理虚拟机器人，取名“阿里小蜜”，阿里小蜜基于客户需求所在的垂直领域（服务、导购、助手等），通过“智能+人工”的方式提供良好的客户体验。

娱乐场景下的聊天机器人系统的主要功能是同用户进行不限定主题的对话（闲聊），从而起到陪伴、慰藉等作用。其应用场景集中在社交媒体、儿童陪伴及娱乐、游戏陪练等领域。有代表性的系统如微软的“小冰”、微信的“小微”、北京龙泉寺的“贤二机器僧”等。其中微软的“小冰”和微信的“小微”除了能够与用户进行开放主题的聊天，还能提供特定主题的服务，如支持用户询问天气、回答用户关于生活常识的疑问等。

应用于**教育场景下的聊天机器人系统**可以根据教育内容的不同进一步划分。例如，通过构建交互式的语言使用环境，帮助用户学习某种语言的聊天机器人；在用户学习某项专业技能时，指导用户逐步深入地学习并掌握该技能的聊天机器人（如前述介绍的 UC 聊天机器人）；在用户的特定年龄阶段，帮助用户进行某种知识的辅助学习的聊天机器人（如目前流行的儿童教育机器人）等。这类聊天机器人的应用场景为具备人机交互功能的学习、培训类产品，以及儿童智能玩具等。

个人助理类应用可以通过语音或文字与用户进行交互，实现用户个人事务的查询及代办，如天气查询、短信收发、定位及路线推荐、闹钟及日程提醒、订餐等，从而让用户可以更便捷地处理日常事务。个人助理的典型应用场景为便携式移动终端设备，如智能手机、智能耳机、笔记本电脑等。

智能问答类聊天机器人系统可以回答用户以自然语言形式提出的事实型问题及其他需要计算和逻辑推理的复杂问题，以满足用户的信息需求并起到辅助用户决策的目的。智能问答聊天机器人的应用场景相对单一，通常作为问答服务整合到聊天机器人系统中。聊天机器人系统不仅要考虑如 What、Who、Which、Where、When 等事实型问答，也要考虑如 How、Why 等非事实型问答，因此智能问答的聊天机器人通常作为聊天机器人的一个服务模块。典型的智能问答系统包括 IBM 研发的 Watson、沃尔夫勒姆研究公司开发的搜索引擎

WolframAlpha[①]、Peak Labs 开发的搜索引擎 Magi[②]等，且后两者都属于基于结构化知识库构建的问答系统。

2．基于实现方式的聊天机器人分类

从实现的角度看，聊天机器人可以分为**检索式**和**生成式**。检索式聊天机器人的回答是提前定义的，在聊天时机器人使用规则引擎、模式匹配或者机器学习训练好的分类器从知识库中挑选一个最佳的回复展示给用户。也就是说，需要事先准备一个知识库，聊天机器人系统接收到用户输入的句子后，在知识库中以检索的方式进行应答内容提取。这种实现方式对知识库的要求相对较高，需要预定义的知识库足够大，尽量多地匹配用户问句，否则检索式聊天机器人系统会经常出现找不到合适回复的情况。这种实现方式的优点是回答的质量高，表达比较自然。生成式聊天机器人则采取不同的技术思路，不依赖于提前定义的回答，但是在训练机器人的过程中，需要大量的语料，语料包含上下文聊天信息和回复。使用这种模型的机器人在接收到用户输入的自然语言后，将采用一定技术手段自动生成一句话作为对用户输入的应答，生成式聊天机器人的优点是可能覆盖任意话题、任意句式的用户输入，缺点是生成的应答句子的质量很可能存在问题，比如出现语句不通顺、句法错误等比较低级的错误。

尽管目前在具体的生产环境中，提供聊天服务的一般都是基于检索的聊天机器人系统，但是基于深度学习的 seq2seq（sequence to sequence）模型的出现可能使基于生成的聊天机器人系统成为主流。

3．基于功能的聊天机器人分类

基于功能的聊天机器人大致可以分为问答系统、面向任务的对话系统、闲聊系统和主动推荐系统 4 种，对这 4 种聊天机器人系统的总结如表 1-1 所示。

① http://www.wolframalpha.com
② http://www.peak-labs.com/#magi

表 1-1　基于功能的聊天机器人分类

分　　类	问答系统	面向任务的对话系统	闲聊系统	主动推荐系统
所属领域	特定领域	特定领域	开放领域	特定领域
主要功能	知识获取	完成用户期望的任务或动作	陪用户闲聊	信息主动推荐
典型应用场景	客服	预订机票	娱乐、情感陪伴	为用户个性化地推荐信息
典型应用	IBM Watson	苹果 Siri	微软小冰	今日头条

目前，对问答系统和主动推荐系统的评价指标较为客观，评价方式也相对成熟。而面向任务的对话系统和闲聊系统，在给定相同输入的情况下，系统回复形式可以多种多样，对于用户的同一输入，通常有多种合理且数目不固定的回复，这使得很难通过一种客观的机制对其进行评价，所以在评价时需要加入人的主观判断作为评价的依据之一。

本书将按照功能分类依次介绍问答系统、面向任务的对话系统、闲聊系统的实现和测评。

1.3　聊天机器人生态介绍

通常，一个完整的聊天机器人系统的框架如图 1-3 所示，其主要包含自动语音识别、自然语言理解、对话管理、自然语言生成、语音合成 5 个主要的功能模块。需要特别指出的是，并不是所有的聊天机器人系统都需要语音技术。例如，以文字方式实现人机交互的聊天机器人系统，就不需要自动语音识别模块和语音合成模块。因此，自动语音识别和语音合成并不作为本书介绍的重点，笔者将对这两部分技术在聊天机器人系统中的地位和作用进行简单介绍。

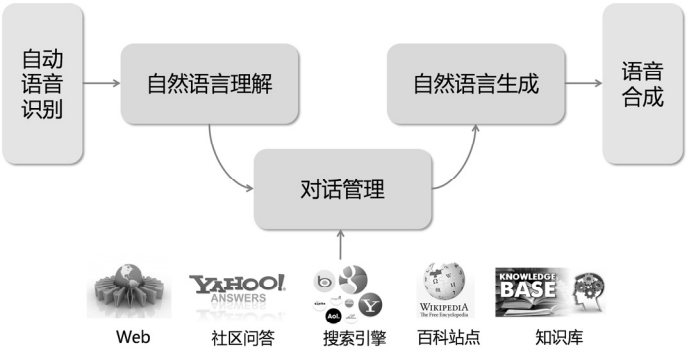

图 1-3　聊天机器人系统的框架

（1）自动语音识别（Automatic Speech Recognition，ASR）模块负责将原始的语音信号转换成文本信息。

（2）自然语言理解（Natural Language Understanding，NLU）模块负责将识别到的文本信息转换为机器可以理解的语义表示。

（3）对话管理（Dialogue Management，DM）模块负责基于当前对话的状态判断系统应该采取怎样的动作。

（4）自然语言生成（Natural Language Generation，NLG）模块负责将系统动作/系统回复转变成自然语言文本。

（5）语音合成（Text-to-Speech，TTS）模块负责将自然语言文本变成语音信号输出给用户。

图 1-4 给出了聊天机器人的生态体系。围绕聊天机器人生态圈，从产品的角度分析，除了有硬件形态的 Amazon Echo、公子小白等聊天机器人，还有纯软件的如苹果的 Siri 和微软的小冰等。为了加速聊天机器人的研发，2016 年前后，不少巨头或创业企业开始对外提供聊天机器人框架（Bot Framework），以 SDK 或 SaaS 服务的形式向第三方公司或个人开发者提供可以用于构建特定应用和领域的聊天机器人，典型代表包括 Amazon Alexa 使用的 Amazon Lex 服务、

微软推出的包含在认知服务（Cognitive Services）大框架下的 LUIS with Bot、api.ai［后被谷歌（Google）收购］、Wit.ai（后被 Facebook 收购）等。除了提供开发聊天机器人的 API，许多聊天机器人平台（Bot Platform）已经在考虑如何将其开发的聊天机器人系统部署到一些常用平台（如微信或 Facebook 等）。

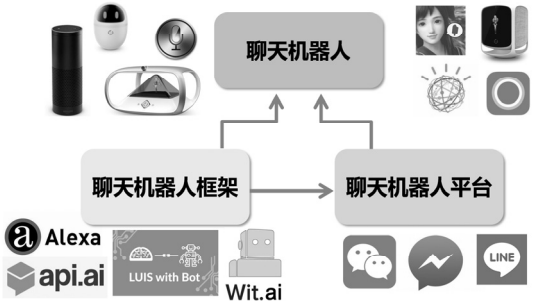

图 1-4　聊天机器人的生态体系

笔者将分别介绍聊天机器人体系下，框架、平台和产品的典型代表。

1.3.1　典型聊天机器人框架介绍

Amazon Lex 是一种可以在任何程序中使用语音和文本构建对话界面的服务。Amazon Lex 具有高级自动语音识别功能，可以将语音转换为文本，还提供自然语言理解功能，用以识别文本的意图，让开发者能够快速构建极具用户吸引力且会话交互高度拟人的应用程序。借助 Amazon Lex，Amazon 将支持 Amazon Alexa 的深度学习技术提供给所有开发人员使用，从而使开发人员能够轻松快速地构建出具备自然语言理解能力的精密对话机器人，即可以支持更深层次人机交互的聊天机器人。

Amazon Lex 提供可扩展、安全且易于使用的端到端（end2end）解决方案，以构建、发布和监控开发人员发布的机器人。图 1-5 形象化地展示了聊天机器人如何通过对话的方式协助用户完成订花的需求。

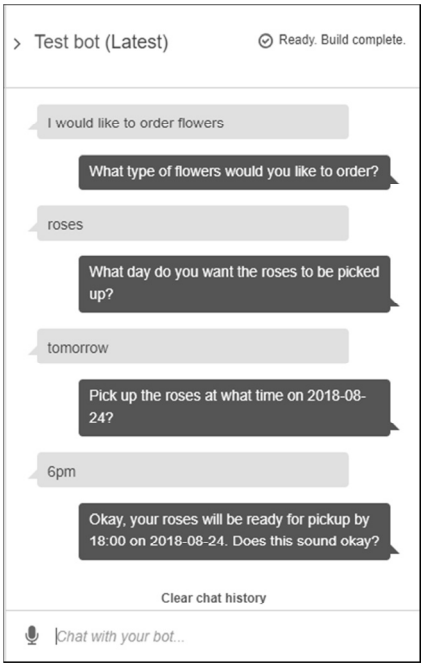

图 1-5　聊天机器人在订花场景的应用

另一个典型的聊天机器人框架是 Facebook 的 Wit.ai。Wit.ai 积累了大量高质量的对话数据，有效促进了聊天机器人系统的发展，并通过将人工智能和人类智能结合，进一步提升了聊天机器人的智能水平，图 1-6 所示为 Wit.ai 网站的截图。

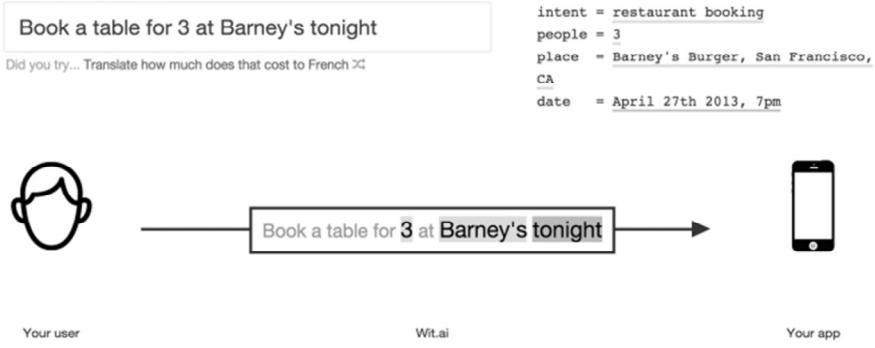

图 1-6　Wit.ai 网站的截图

1.3.2 聊天机器人平台介绍

开发人员可以通过聊天机器人“框架”构建自己的聊天机器人“产品”，进而将自己的产品部署在合适的聊天机器人“平台”上。例如，微信公众平台可以被看作聊天机器人平台，各种服务机构和个人可以利用微信公众平台开发和部署面向自己服务对象的聊天机器人，满足用户的要求。我们熟悉的微信个人号、微信群、微信公众号和微信服务号的很多功能（例如自动回复等）都是由虚拟的聊天机器人实现的。微信也通过对运营者开放接口，允许他们接入采用第三方服务设计的聊天机器人，进一步促进了聊天机器人开发和应用的迅速增长。

另一个具有代表性的平台是小 i 机器人，其提供智能机器人技术和平台，建立了包括知识表示、推理预测、机器学习（深度学习）、语义理解、分析决策，以及聊天机器人开发的完整架构，并同时拥有商业化的智能机器人应用产品。

1.3.3 典型的聊天机器人产品介绍

我们已经介绍了聊天机器人的 4 种分类，包括问答系统、面向任务的对话系统、闲聊系统和主动推荐系统，本节选取有代表性的聊天机器人产品详细介绍。

1. 苹果公司发布的个人语音助理 Siri

2011 年 10 月 14 日，苹果公司在其 iPhone 4S 发布会上推出了新一代智能个人助理 Siri。Siri 具备聊天和执行用户指令的功能，苹果公司将其视为移动终端应用的总入口。但是，由于系统本身自动语音识别能力、自然语言理解能力的不足，以及受到用户习惯使用语音和 UI（User Interface，用户界面）操作两种形式进行人机交互等问题的限制，Siri 没能真正担负起个人事务助理的重任。无论如何，Siri 通过自然语言交互的形式实现问答、推荐、手机操作等功能，并被集成进 iOS 5 及之后的 iOS 版本中，使得个人智能助理的概念在用户群体中得到广泛认知。Siri 被定位为面向任务的对话系统，为用户提供打电话、订餐、订票、放音乐等服务。这些服务本质上都是使用指令指示手机操作系统去

完成一个任务，指令的产生需要 Siri 理解用户的输入和意图，因此会涉及自然语言理解。Siri 对接了很多服务，且设置了“兜底”操作，当 Siri 无法理解用户的输入时就命令搜索引擎返回相关的服务。Siri 的出现引领了移动终端个人事务助理应用的商业化发展潮流。

图 1-7 给出了 Siri 的技术框架，其主要流程包括自动语音识别、自然语言理解（既包括分词、词性标注、命名实体识别等基础自然语言处理技术，也包括模式动作映射等语义解析模块）、服务管理、回复生成，以及语音合成等 5 个部分，其中，服务管理整合了内部和外部的各类服务接口，例如电子邮件、地图、天气等，从而给用户提供了更便捷的语音助手服务。

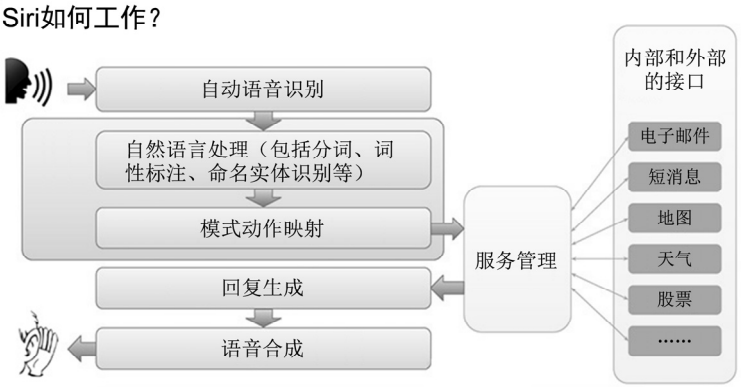

图 1-7　Siri 技术框架介绍

2．IBM 公司发布的“最强大脑”

2011 年 2 月，IBM 耗资 3000 万美元研发的 IBM Watson 登上了美国著名智力问答竞赛节目《危险边缘》（*Jeopardy*），面对节目中充满双关意思的英文问题，IBM Watson 能做出分析并在庞大的自然语言知识库中寻找线索，将这些线索组合成答案。最终，IBM Watson 压倒性地击败了节目中最聪明的人脑，同时创下了这个知识竞赛系列节目 27 年历史上的最高分。IBM Watson 作为 IBM 公司研发的问答系统，集成了自然语言处理、信息检索、知识表示、自动

推理、机器学习等多项技术的应用，形成了假设认知和大规模的证据搜集、分析、评价的深度问答技术。IBM Watson 可以分析自然语言形式的数据，通过大规模学习和推理，为用户提供个性化服务。

本书后面章节将对 Watson 的实现方法进行详细介绍。

3．谷歌公司发布的智能个人助理 Google Now

2012 年 7 月 9 日，谷歌发布了智能个人助理 Google Now。Google Now 通过自然语言交互的方式为用户提供页面搜索、自动指令等功能。谷歌最大的优势是强账号关联，通过 Gmail 将不同平台的账号连接在一起，针对具体用户实现了非常强的个性化。举例来说，谷歌为 Gmail 做了一个智能回复（smart reply）功能，根据用户的习惯、风格帮用户形成模板。此处，笔者顺带介绍谷歌公司的 Allo（如图 1-8 所示），Allo 是谷歌在前述工作的基础上发布的语音助手。Allo 具备随时间推移学习用户行为的能力。

图 1-8　谷歌公司的 Allo

4．微软发布的个人机器人助理 Cortana 和聊天机器人小冰

2014 年 4 月 2 日，微软发布个人机器人助理 Cortana。微软将 Cortana 定位为个人助理，并将其嵌入微软公司发布的 Windows 操作系统中。同时，微软发布了另一款聊天机器人——小冰，它主要用于闲聊和情感陪伴。2018 年 7 月，

微软对小冰进行了功能升级，推出了第六代小冰[①]。

图 1-9 所示为微软官网对小冰的介绍，图 1-10 所示为小冰的对话示例。

图 1-9　微软官网对小冰的介绍

作为聊天机器人的一种分类，主动推荐系统采用的是一种实现个性化信息推送的技术方式。主动推荐系统并不需要用户提供明确的需求，而是通过分析用户的历史行为数据建立用户画像，从而基于用户画像主动向用户推荐系统认为能够满足用户兴趣和需求的信息。在电商购物（如阿里巴巴、亚马逊）、社交网络（如 Facebook、微博）、新闻资讯（如今日头条）、音乐电影（如网易云音乐、豆瓣）等领域均有广泛而成功的应用。主动推荐系统本质上是一项帮助人们解决信息过载（information overload）问题的工具。所谓信息过载，是指用户真正需要、真正感兴趣的东西被淹没在其同类物品的海洋里。为了找到它，用户需要耗费大量的时间和精力。为了解决信息过载问题，迄今为止我们经历了分类目录、搜索引擎、主动推荐系统 3 个技术发展阶段。搜索引擎可以满足

① http://www.msxiaoice.com/

用户的明确目的、主动查找的需求，主动推荐系统能够在用户没有明确目的时帮助用户发现他们感兴趣的新内容。传统的推荐系统一般仅考虑用户对推荐对象的评分（User-item Rating），不考虑时间、地点、场景、情绪、活动状态等上下文，无法适应相对复杂的环境；而主动推荐系统一般会考虑更多的上下文场景，通过结合更丰富、多维度的用户画像信息，向用户提供更准确、更有效的推荐，实现聊天机器人的主动交互。

图 1-10　与小冰的聊天示意图

图 1-11 展示了一个典型的由用户主导的对话过程，机器人只是单纯地回答用户提出的问题。图 1-12 展示了一个机器人主动和用户交互的对话过程，机器人理解了“用户可能感冒了”这一隐藏的事实之后对用户进行了相应的关怀。通过这个小例子可以发现，主动的交互方式能够显著提升用户体验，且机器人主动交互的方式更接近真实的人与人之间的对话方式，使得对话更自然。

图 1-11　由用户主导的对话过程

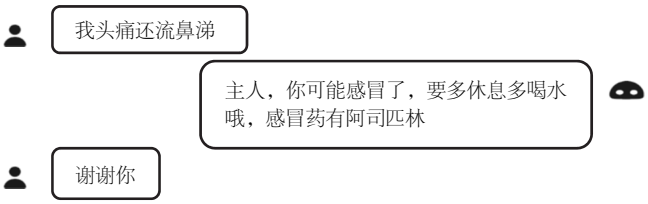

图 1-12　机器人主动推荐的对话

一种实现主动推荐的方式，是基于知识图谱（Knowledge Graph）的主动推荐系统。例如，在建立音乐领域的主动推荐系统时，可以先建立音乐领域知识图谱和用户知识图谱，然后在进行用户信息搜集的过程中建立起用户的音乐喜好画像，从而更精准地对用户进行音乐推送。利用微信公众号实现主动推荐系统的流程如图 1-13 所示。

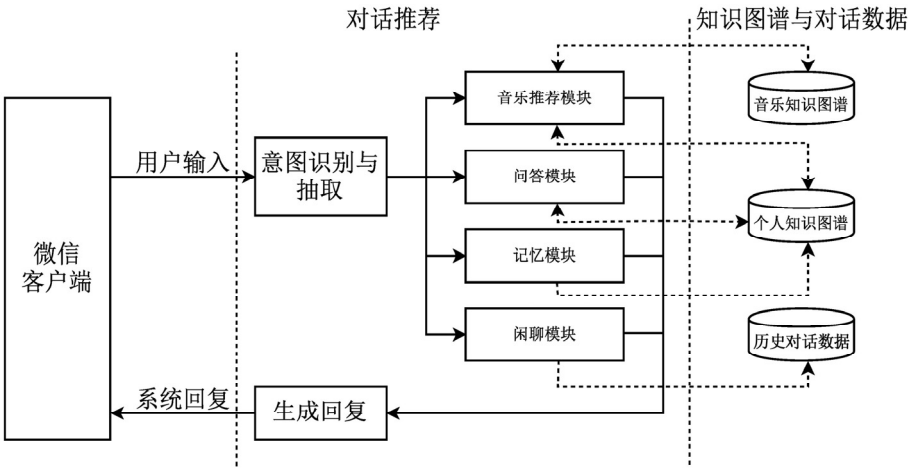

图 1-13　利用微信公众号实现主动推荐系统的流程

从图 1-13 中可以看出，在用户点播歌曲的过程中，主动推荐系统可以结合音乐知识图谱、用户的个人知识图谱，以及用户的历史对话数据，综合给出最优的音乐推荐。

主动推荐系统与问答系统、面向任务的对话系统和闲聊系统被认为是聊天机器人产品的 4 种主要分类，但由于主动推荐系统的发展时日尚短，从技术到产品逻辑都不甚完善，在后续具体介绍不同类别聊天机器人产品使用的具体技术时，就不再将主动推荐系统作为单独章节进行介绍。

1.4 参考文献

[1] J. Weizenbaum, ELIZA—a computer program for the study of natural language communication between man and machine, Communications of the ACM, vol. 9, No. 1, pp. 36-45, 1966.

2 聊天机器人技术原理

在正式介绍聊天机器人关键技术之前，对不同类型聊天机器人系统的技术侧重点进行说明，以便读者更好地理解聊天机器人的关键技术。

目前，较流行的聊天机器人系统包括问答系统、面向任务的对话系统、闲聊系统，以及新近流行的主动推荐系统。

1．问答系统

问答系统（Question Answering，QA）由最初的搜索需求发展而来，基本为“一问一答”的交互模式，因此构建问答系统时一般不会涉及对话管理相关的技术。第 1 章介绍过，聊天机器人的核心模块包括自然语言理解、对话管理和自然语言生成。在自然语言理解层面，问答系统偏重于问句分析，旨在获取问句的主题词、问题词、中心动词等。目前，问句分析主要采用**模板匹配**和**语义解析**两种方式。

2．面向任务的对话系统

面向任务的对话系统，其目的是解决用户的明确需求。面向任务的对话系统通过对话管理跟踪当前的对话状态，进而明确用户的目的和需求，因此，对话管理是面向任务的对话系统的一个技术侧重点。也就是说，不同于问答系统不涉及对话管理技术的情况，**对话管理**在面向任务的对话系统中**占据重要位置**。面向任务的对话系统中的自然语言理解技术并不侧重于对某类句子和某类词的识别，而是聚焦于将用户输入的自然语言映射为用户的意图和相应的槽位值（意图和槽位的定义详见本书 4.2 节）。

3．闲聊系统

闲聊系统针对的是用户没有特定目的、没有具体需求情况下的多轮人机对话，其构造过程中需要同时注意**对话管理**（上下文多轮交互）和**自然语言理解**两个模块的构建。

4．主动推荐系统

主动推荐系统仍处于起步阶段，作为人机自然交互的关键一环，其作用更多是体现聊天机器人的认知能力。

从技术的角度看，不同类型的人机对话系统都包括**自然语言理解**、**自然语言生成**和**对话管理** 3 个模块，但不同类型的人机对话系统偏重的技术模块及在各个模块中使用的技术细节均有所不同。因此，本章将从这 3 个层面阐述搭建聊天机器人的通用技术。

2.1　自然语言理解

笔者先对自然语言的出现和作用进行简述。

语言是指生物同类之间由于沟通需要而制定的指令系统，语言与逻辑相关，目前只有人类才能使用体系完整的语言进行沟通和思想交流。

自然语言通常会自然地随文化发生演化，英语、汉语、日语都是具体种类的自然语言，这些自然语言履行着语言最原始的作用：人们进行交互和思想交流的媒介性工具。我们可以从语音、音韵、词态、句法、语义、语用 6 个维度理解自然语言。

（1）**语音**是与发音相关的学问（例如儿童学习的汉语拼音等），主要在前述介绍的语音技术中发挥作用。

（2）**音韵**是由语音组合起来的读音，即汉语拼音和四声调。

（3）**词态**封装了可用于自然语言理解的有用信息，其中信息量的大小取决于具体的语言种类。需要特别提及的是，中文没有太多的词态变换（不像拉丁语系语言），仅存在不同的偏旁，导致出现词的性别转换的情况（例如“他”“她”）。

（4）**句法**主要研究词语如何组成合乎语法的句子，句法提供单词组成句子的约束条件，为语义的合成提供框架。

（5）**语义**和**语用**是自然语言所包含和表达的意思。

对计算机来说，自然语言处理的难度主要体现在以下几个方面：

（1）自然语言千变万化，没有固定格式。同样的意思可以使用多种句式来表达，同样的句子调整一个字、调整语调或者调整语序，表达的意思可能相差甚多。

（2）不断有新的词汇出现，计算机需要不断学习新的词汇。

（3）在不同的场景（上下文语境）下，同一句话表达的意思可能不同。

2.1.1　自然语言理解概览

1. 什么是自然语言理解

自然语言理解以语言学为基础，融合逻辑学、计算机科学等学科，通过对语法、语义、语用的分析，获取自然语言的语义表示，其目的是为聊天机器人生成一种机器可读的自然语言的语义表示形式，图 2-1 形式化地表示了自然语言理解的作用。

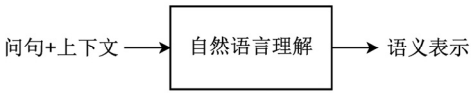

图 2-1　自然语言理解的作用的形式化表示

从图 2-1 中可以看出，自然语言理解模块以自然语言作为输入，处理后输出对应输入的机器可读的语义表示。自然语言理解作为聊天机器人系统的基础核心模块，面临以下几个挑战：

首先，自然语言理解的准确率受语音识别（本书第 1 章已经对语音识别相关内容进行了介绍）准确率的影响。目前语音识别的错误率在复杂环境下仍然较高，这会对自然语言理解的准确性产生负面影响。

其次，自然语言所表达的语义本身存在一定的不确定性，同一句话在不同语境下的语义可能完全不同。例如，用户说的“肯德基到家”这句话，可能是外卖需求，也可能是从肯德基到用户家的叫车需求，需要配合上下文语境才能更好地理解这类句子表达的含义。再比如，用户说“明早 8 点叫我起床”，如果用户输入发生在 23:00，那么“明早”指的是“第二天早上”；如果用户输入发生在 00:01，那么“明早”很有可能指的就是“当天早上”。

最后，人类讲话时往往出现不流畅、错误、重复等情况，而对机器来说，在它理解一句话时，这句话中每个词的确切含义并不重要，重要的是这句话整体所表达的意思。

在中文自然语言理解中还有一点需要特别注意：英文文本中单词之间是以空格作为自然分界符的，而中文文本中词语并没有形式上的自然分界符，因此在面向中文的聊天机器人系统中需要先对文本进行分词处理。除此之外，词性是词汇的一个很强的特征，因此，通常在分词的同时还要对单词进行词性标注。

2．聊天机器人中的自然语言理解

聊天机器人系统中的自然语言理解模块的功能主要包括实体识别、用户意图识别、用户情感识别、指代消解、省略恢复、回复确认及拒识判断等。**实体识别**又称命名实体识别（Named Entity Recognition），指识别自然语言中具有特定意义的实体，如人名、时间、地名及各种专有名词。**用户意图识别**中需要识别的用户意图包括**显式意图**和**隐式意图**，显式意图通常对应一个明确的用户需求，而隐式意图则较难判断，表 2-1 举例说明了用户的显式意图和隐式意图。用户情感和用户意图类似，也可以分为显式和隐式两种，表 2-2 是对用户显式情感和隐式情感的举例说明。**指代消解**和**省略恢复**是指聊天主题背景一致的情况下，人们在对话过程中通常会习惯性地使用代词指代已经出现过的某个实体或事件，或者为了方便表述省略句子部分成分的情况。自然语言理解模块需要明确代词指代的成分及句子中省略的成分，唯有如此，聊天机器人才能正确理解用户的输入，给出合乎上下文语义的回复。当用户意图、聊天信息等带有一定的模糊性时，需要聊天机器人主动向用户询问，确认用户的意图，即**回复确认**。**拒识判断**是指聊天机器人系统应当具备一定的拒识能力，主动拒绝识别及回复超出自身理解/回复范围或者涉及敏感话题的用户输入。

表 2-1　用户显式意图和隐式意图的例子

用户显式意图示例	用户隐式意图示例
用户：明天帮我预订蛋糕	用户：好热啊
意图：Book_Cake	意图：Set_Ac 或 Get_Temp
说明：明确表示了用户预订蛋糕的意图	说明：可能想知道当前气温或控制空调

表 2-2　用户显式情感和隐式情感的例子

用户显式情感示例	用户隐式情感示例
用户：今天心情真好	用户：今天和客户谈判出问题了
情绪：正面	情绪：负面
说明：明确表示了喜悦的心情	说明：计算机不太容易判断用户此时的情感

3．自然语言理解技术概述

通常，可以将自然语言理解的主要方法分为基于规则的方法和基于统计的方法两种。**基于规则的方法**是指利用规则定义如何从文本中提取语义，大致思路是人工定义很多语法规则，它们是表达某种特定语义的具体方式，然后自然语言理解模块根据这些规则解析输入该模块的文本。基于规则的自然语言理解模块的优点是灵活，可以定义各种各样的规则，而且不依赖训练数据；缺点是需要大量的、覆盖不同场景的规则，且随着规则数量的增长，对规则进行人工维护的难度也会增加。因此，**基于规则的自然语言理解只适合用在相对简单的场景，其优势在于可以快速实现一个简单可用的语义理解模块。**

当数据积累到一定程度，就需要考虑使用基于统计的自然语言理解方法。**基于统计的自然语言理解方法**通常使用大量的数据训练模型，并使用训练所得的模型执行各种上层语义任务。其优点是数据驱动且健壮性较好，缺点是训练数据难以获得且模型难以解释和调参。基于统计的自然语言理解通常使用数据驱动的方法解决分类和序列标注问题，因此研究人员将意图识别定义成一个分类问题。这个问题的输入是句子的文本特征，输出是句子文本特征所属的意图分类，SVM、AdaBoost 算法等可以被用来解决该问题。实体抽取则可以被直观地描述成一个序列标注问题，该问题的输入是句子的文本特征，输出是文本特征中的每个词或每个字属于某一实体的概率，用隐马尔可夫模型（Hidden Markov Model，HMM）、条件随机场（Conditional Random Field，CRF）等算法可以有效地解决该问题。另外，当数据量足够大时，使用基于神经网络的深度学习方法处理意图识别和实体抽取任务可以取得更好的效果。

与基于规则的自然语言理解相比，基于统计的方法靠数据驱动，数据量越大，数据的分布就越能表征真实情况，模型效果的健壮性就越好。基于上述描述，我们可以发现，基于统计的方法需要训练数据，尤其是如果使用深度学习方法，更是需要大量的数据。同时，由于长尾数据（即出现次数较少的数据场景）普遍存在，基于统计的方法在实际应用中的效果也受训练数据质量的影响。

在具体实践中，通常将这两种方法结合起来使用。

（1）没有数据及数据较少时先采用基于规则的方法，当数据积累到一定规模时逐渐转为使用基于统计的方法。

（2）基于统计的方法可以覆盖绝大多数场景，在一些基于统计的方法覆盖不到的场景中使用基于规则的方法兜底，以此来保证自然语言理解的效果。

可以说，自然语言理解是所有聊天机器人系统的基础，目前许多公司将自然语言理解作为一种云服务提供，方便其他产品快速地具备语义理解能力。例如，Facebook 的 Wit.ai、谷歌的 api.ai 和微软的 LUIS.AI 都是类似的服务平台。具体来说，使用者需要上传数据到服务平台，服务平台根据数据训练出模型，并提供训练所得模型的接口供使用者调用。使用这类服务平台能够快速地搭建出数据驱动的自然语言理解模块，但这些服务平台过度强调通用性，且数据处理方式和业务处理逻辑对普通开发者来说都是黑盒，因此开发者的定制化需求很难被满足。

2.1.2 自然语言理解基本技术

词法分析、句法分析、语义分析等基本的自然语言处理技术对聊天机器人系统中的自然语言理解功能起到了至关重要的作用，三者的关系可以用图 2-2 大致表示。

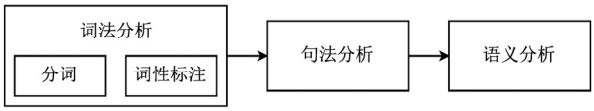

图 2-2　词法分析、句法分析、语义分析的关系

1．词法分析

汉语具有大字符集（常用汉字约有六七千字，远远多于英文的 26 个字母）、词与词之间没有明确分隔标记、多音现象严重、缺少形态变化（单复数、时态、阴阳性）等特点。这些特点给汉语词法分析带来了分词词表的建立、重叠词区分（如“黑”和“黑黑的”）、歧义字段切分、专有名词识别等问题。

从学术的角度看，词是语言中能独立运用的最小单位，也是信息处理的基本单位，汉语的词法分析（Lexical Analysis）包括汉语分词和词性标注两部分。

中文不同于英文，其没有自然分隔符（明显的空格标记），因此汉语自然语言处理的首要工作就是将输入的字串切分为单独的词语，这一步称为分词（Word Segmentation）。目前采用的汉语分词方法主要有基于词表匹配的方法和基于统计模型的方法。

基于词表的方法，会逐字对字符串进行扫描，发现字符串的子串和词表中的词相同就算匹配。基于词表的方法通常有正向最大匹配法、逆向最大匹配法、双向扫描法和逐词遍历法等。常见的基于词表的分词工具有 IKAnalyzer、庖丁解牛等。**基于统计模型的方法**根据人工标注的词性和统计特征对中文进行建模，通过模型计算各种分词出现的概率，将概率最大的分词结果作为最终结果。基于统计模型方法的常用算法为 HMM、CRF 等。常见的基于统计模型的分词工具有 ICTCLAS、Stanford Word Segmenter 等。深度学习兴起后，长短期记忆网络（LSTM）结合 CRF 的方法得到了快速发展。

词性是词语最基础的语法属性之一，因此研究者通常将词性标注（Part-Of-Speech Tagging，POS Tagging）看作词法分析的一部分。

词性标注的目的是为句子中的每个词赋予一个特定的类别，即为分词结果中的每个单词标注词性（例如名词、动词、介词等都是单词的词性）。这个过程是非常典型的序列标注问题。一个句子中最重要、最能体现句子所包含信息的 4 种词性为名词、动词、形容词和副词，这 4 类词属于开放类型(Open Class)，其中的词量会随时间增加；相对于开放类型，封闭类型（Closed Class）中的词相对固定，其中包括冠词、介词、连词等。词性标注是根据词的功能将词分组的典型方法，表 2-3 针对自然语言“The results appear in today's news”给出了一个特定的词性标注示例。

表 2-3　对句子进行词性标注的示例

词	The	results	appear	in	today	's	news
词性	det	noun	verb	preposition	noun	possessive	noun

词性标注最初采用的主要模型是隐马尔可夫生成式模型，之后陆续采用过判别式的最大熵模型、支持向量机模型等进行尝试。词性标注的方法主要分为两种：基于规则的方法和基于统计模型的方法。**基于规则的词性标注方法**按照兼类词搭配关系和上下文语境建造词类消歧规则。**基于统计模型的词性标注方法**通过模型计算各种词性出现的概率，将概率最大的词性作为最终结果。学术界通常采用结构感知器模型和条件随机场模型解决词性标注问题。随着深度学习技术的发展，研究者也提出了很多行之有效的基于深层神经网络的词性标注方法。词性标注常用的工具有 Stanford Log-linear Part-Of-Speech Tagger、哈工大的 LTP 工具等。

词性标注近几年的主要进展集中在词性标注和句法分析联合建模、异构数据融合和基于深度学习的标注方法上。词性标注和句法分析紧密相关，因此联合建模可以同时提高词性标注和句法分析两个任务的准确率。由于标注规范的不同，目前汉语数据集属于多元异构数据集，如何利用异构数据提升模型准确度也得到了学者的关注。另外，深度学习的发展进一步提升了词性标注的准确

度，典型的方式包括双向 LSTM 结合 CRF 等。

2．句法分析

句法分析（Syntactic Parsing）的主要任务是对输入的文本句子进行分析以得到句子句法结构（Syntactic Structure）。对自然语言的句法结构进行分析，一方面是自然语言理解任务自身的需求，另一方面可以为其他自然语言处理任务提供支持。例如，基于句法驱动和统计的机器翻译需要对源语言、目标语言，或者同时对这两种语言进行句法分析；对自然语言包含的语义进行分析时，通常以句法分析的结果作为语义分析的输入，以便从中获得更多的语义指示信息[①]。

简单来说，句法分析是从字符串得到句法结构的过程。不同的语法形式对应的句法分析算法不同，短语结构和依存结构是目前句法分析中研究最广泛的两类文法体系。由于短语结构语法（特别是上下文无关的语法）应用范围最广，以短语结构树为目标的句法分析器的研究进展最引人瞩目，很多其他形式的语法对应的句法分析器都可以通过对短语结构语法的句法分析器进行简单改造得到。

对句子进行句法分析需要确定句子的句法结构，分析的结果往往以树结构的形式表现，这棵表示句子结构的树又叫作**句法分析树**。句法分析树的建立可以采用自顶向下的方法，也可以采用自底向上的方法。

根据句法结构的不同表示形式，可以将句法分析任务划分为以下 3 种。

（1）依存句法分析（Dependency Syntactic Parsing），主要任务是识别句子中词汇之间的相互依存关系。

（2）短语结构句法分析（Phrase-structure Syntactic Parsing），也称作成分句法分析（Constituent Syntactic Parsing），主要任务是识别句子中短语结构和短语之间的层次句法关系。

① 中国中文信息学会《中文信息处理发展报告（2016）》

（3）深层文法句法分析，主要任务是利用深层文法，对句子进行深层的句法及语义分析，这些深层文法包括词汇化树邻接文法、词汇功能文法、组合范畴文法等。

1）句法分析之依存句法分析

依存句法分析的基本假设是：一个句子中存在主体（被修饰词）和修饰词，句子中词的修饰关系具有方向性，通常是一个词支配另一个词，这种支配与被支配的关系就是**依存文法**，词和词之间的依存（修饰）关系本质上包含在句法结构中。一个依存关系连接的两个词分别是核心词（head）和依存词（dependent），图 2-3 所示为基于图的依存句法分析的一个示例。法国语言学家 L.Tesniere 于 1959 年在著作《结构句法基础》中提出依存句法分析的基本假设，该假设对语言学的发展产生了深远的影响。依存句法分析通过分析语言单位成分之间的依存关系揭示语言的句法结构。依存句法分析理论主张句子中核心动词是支配其他成分的中心成分，而核心动词本身不受其他任何成分支配，所有受支配成分都以某种依存关系的形式从属于其支配者。20 世纪 70 年代，Robinson 提出依存句法中关于依存关系的 4 条公理，中国学者在处理中文信息研究的过程中，基于上述 4 条公理提出了依存关系的第 5 条公理。中文依存类型主要包括 1 个核心类型、18 个补充类型和 14 个辅助类型。

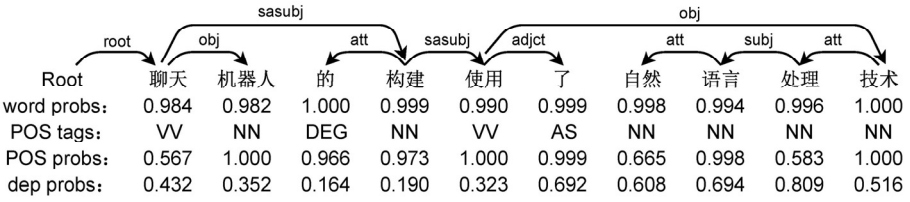

图 2-3　依存句法分析示例

依存关系的 5 条公理如下：

（1）一个句子中只有一个成分是独立的。

（2）其他成分直接依存于某一个成分。

（3）任何一个成分都不能依存于两个或两个以上的成分。

（4）如果 A 成分直接依存于 B 成分，而 C 成分在句中位于 A 和 B 之间，那么 C 或者直接依存于 B，或者直接依存于 A 和 B 之间的某一成分。

（5）中心成分左右两边的其他成分相互不发生关系。

表 2-4 给出了依存分析中常用的关系。需要读者特别注意的是，尽管常用的依存关系的类别相对固定，但同样的依存关系，在不同文章中的标签可能不同。表 2-4 中独立结构的标签为“IS”，也有研究人员将其标记为“S”。

表 2-4　依存分析中常用的关系

关系类型	标　签	描　述	例　子
主谓关系	SBV	subject-verb	他邀请我跳舞（他←邀请）
动宾关系	VOB	直接宾语，verb-object	妈妈给我一个吻（给→吻）
间宾关系	IOB	间接宾语，indirect-object	妈妈给我一个吻（给→我）
前置宾语	FOB	前置宾语，fronting-object	莫我肯顾（我←顾）
兼语	DBL	double	他邀请我跳舞（邀请→我）
定中关系	ATT	attribute	红宝石（红←宝石）
状中结构	ADV	adverbial	特别严厉（特别←严厉）
动补结构	CMP	complement	打扫完了卫生（打扫→完）
同位语	APS	appositive	我本人非常高兴（我←本人）
并列关系	COO	coordinate	天空和海洋（天空→海洋）
介宾关系	POB	preposition-object	在阳光下（在→下）
左附加关系	LAD	left adjunct	天空和海洋（和←海洋）
右附加关系	RAD	right adjunct	朋友们（朋友→们）
独立结构	IS	independent structure	我五岁，他四岁（两个单句在结构上彼此独立）
核心关系	HED	head	美丽的花朵争相开放（“花朵”是整个句子的核心）

目前对依存句法分析的研究主要集中在数据驱动的依存句法分析，即将已有数据集分为训练集和测试集，基于训练集训练得到依存句法分析器，这种方法不涉及对依存语法理论的研究。数据驱动的依存句法方法的主要优势在于只要给定较大规模的训练数据，不需要过多的人工干预就可以得到比较好的依存句法分析器模型。因此，这类方法很容易应用到新领域乃至新的语言环境中。数据驱动的依存句法分析方法主要有两种：基于图（graph-based）的分析方法和基于转移（transition-based）的分析方法。

基于图的分析方法是将字符串（句子）和相应的依存树组成的数据对作为训练数据，训练的目标是学习一个可以预测一句依存树未知的句子的最佳依存树（预测句子对应的最佳图），在建模的过程中需要增加对依存树的限制条件，例如图中的边是有向边，在有向的路径上一个词只能被访问一次，每个词只能有一个支配节点等。在具体操作时，可以基于最大生成树的思想，对所有可能的边进行打分，然后选择分数最多的树。由于特征提取（分数计算）是在每条具体的边上进行的，为了更好地考虑全局特征，可以在特征提取函数中考虑跨越几条边的子图，然后使用动态规划和近似算法增加解码 / 预测的效率。

由于图中的节点和边的无序性，以及预测图的巨大计算开销，基于转移的分析方法得到了应用。基于转移的分析方法本质上将图预测转化为了序列标注问题，主要的转移系统有 arg-eager 系统、arc-standard 系统、easy-first 系统等。这些基于转移的系统主要包含的操作有以下三种：

（1）将单词从缓存（buffer）移入栈（stack）中，将单词从栈中移回。

（2）从栈中将单词弹出（pop）。

（3）创建带有标签（label）的有向边（左向边或右向边）。

总结：对比基于转移的依存分析和基于图的依存分析这两种方式，基于转

移的依存分析在短句上表现较好，这是由于其采用贪心策略，但是在长句的依存分析中容易受到早期错误的影响；而基于图的依存分析在长句依存上有较好的表现，但缺乏丰富的结构化特征。

2）句法分析之短语结构句法分析

短语结构句法分析的研究主要基于上下文无关文法（Context Free Grammar，CFG），短语结构句法分析方法的规则来源可以分为**人工编写规则**和**数据驱动的自动学习规则**两类。人工书写规则的方法的缺点是规则之间的冲突会随规则数量的增多而加剧，为继续添加新规则带来困难。与人工书写规则相比，数据驱动的自动学习规则的方法开发周期短且规则的健壮性强，已经成为短语结构句法分析中的主流方法。为了在句法分析中引入统计信息，提高系统的鲁棒性，通常需要将上下文无关文法扩展为概率上下文无关文法（Probabilistic Context Free Grammar，PCFG），即需要为每条文法规则指定出现的概率值。然后利用最大似然估计（Maximum Likelihood Estimation，MLE）计算每条规则的概率值，是获得概率上下文无关文法最简单、直观的方法。上述方法的实现比较简单，但由于上下文无关文法采取的独立性假设过于严格（独立性假设的内容为：一条文法规则的确定仅与该规则左侧句子中的非终结符有关，与其他上下文信息无关），导致文法中缺乏其他信息用于规则消歧，因此所建立分析器的性能较低。针对上述问题，研究人员先后提出了两种弱化上下文无关假设的改进思路：一种思路是使用词汇化（Lexicalization）的方法，在上下文无关文法规则中引入词汇信息；另一种思路是使用符号重标记（Symbol Refinement）的方法，通过改写（细化或者泛化）非终结符的方式将上下文信息引入句法分析器。

3）句法分析之深层文法句法分析

与前两种句法分析方法不同，深层文法句法分析相关的研究相对较少。词汇化树邻接文法（Lexicalized Tree Adjoining Grammar，LTAG）、词汇功能文

法（Lexical Functional Grammar，LFG）和组合范畴文法（Combinatory Categorial Grammar，CCG）是深层文法句法分析中较成熟的 3 种方法。

从本质上讲，依存句法分析是浅层句法分析，其实现过程相对简单，可以提供的信息也相对较少，比较适合应用在多语言环境中。深层文法句法分析采用的文法相对复杂，提供的句法和语义信息也较为丰富，复杂的文法提高了分析器运行的复杂度，为深层句法文法分析处理大规模数据带来了难度。

除了上述 3 种句法分析方法，深度学习在句法分析中的应用逐渐成为研究热点，研究工作主要集中在特征表示方向。基于传统方法的特征表示主要采用人工定义原子特征和特征组合的方法，而基于深度学习的句法分析方法通过将句子的原子特征向量化，利用多层神经网络提取特征对句子进行表示。也就是说，基于深度学习的句法分析方法首先将句子的词、词性、类别标签等原子特征表示为向量，然后利用多层网络进行特征提取，该过程如图 2-4 所示。

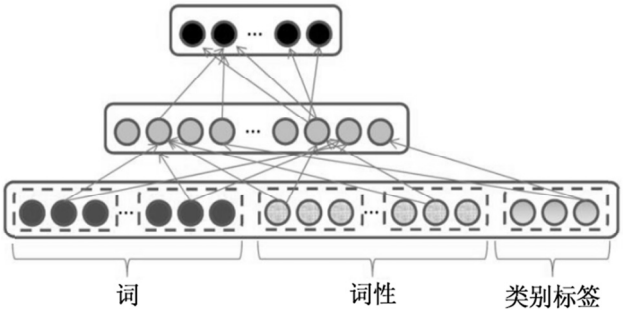

图 2-4　基于深度学习的特征提取

深度学习在特征表示方面有如下优点：

（1）只需要句子的原子特征。在传统的实现方法中，通过人工组合形成一元特征、二元特征、三元特征、四元特征甚至更多元的特征，这种人工组合的方式理论上可以取得较好的效果，然而由于形成最佳特征集合的组合方式未知，导致人工组合方法在应用时难以取得较好的效果。深度学习方法将句子的所有

原子特征向量化，然后采用向量乘法等非线性运算对这些向量化特征进行组合，理论上能实现任意元的特征组合。

（2）能使用更多的原子特征。比如在基于图的模型中建立弧时，深度学习的方法不仅可以使用左边第一个词、右边第一个词等原子特征，还可以使用左边整个词序列、右边整个词序列等更多的特征。研究人员把基于深度学习的特征表示方法分别应用在基于图的句法分析模型和基于转移的句法分析模型中，结果证明了深度学习方法在句法分析中的优势。

3．语义分析

语义，指的是自然语言所包含的意义，在计算机科学领域，可以将语义理解为数据对应的现实世界中的事物所代表概念的含义。语义分析（Semantic Analysis）指运用各种机器学习方法，让机器学习与理解一段文本所表示的语义内容。语义分析是一个非常广的概念，任何对语言的理解都可以归为语义分析的范畴。语义分析涉及语言学、计算语言学、人工智能、机器学习，甚至认知语言等多个学科，是一个典型的多学科交叉研究课题。

语义分析的最终目的是理解句子表达的真实含义。 具体阐述如下：

（1）语义分析在机器翻译任务中有重要应用。在过去 20 多年的发展历史中，统计机器翻译主要经历了基于词、基于短语和基于句法树的翻译模型。通过将语义分析应用于统计机器翻译，可以有效提升机器翻译的性能。

（2）基于语义的搜索一直是搜索追求的目标。语义搜索，是指搜索引擎的工作不再拘泥于根据用户输入搜索关键词的字面意思，还能捕捉到用户所输入关键词背后的真正意图，并以此进行搜索，从而保证向用户返回的是最符合其需求的搜索结果。

（3）语义分析是实现大数据的理解与价值发现的有效手段。语义分析与大数据在某种程度上互为基础。一方面，如果想得到更精确的语义分析结果，需要大数据的支持，即从大数据中挖掘并形成更大、更齐全、更精确的知识库，而知识库对语义分析的性能有重要的影响；另一方面，如果想从大数据库中挖掘出更多、更有用的信息，人们需要用到语义分析等自然语言处理技术。

对聊天机器人系统来说，通过语义分析可以获取用户的意图、情感，并通过对上下文语境的语义建模保持聊天机器人系统的个性一致。

2.1.3 自然语言表示和基于深度学习的自然语言理解

在解决自然语言理解领域的问题时，通常的做法是先将自然语言表示为计算机可以理解的形式。介绍了自然语言理解基本技术后，下面介绍 3 种常用的文本特征表示模型，这 3 种模型往往被用于表示文本或自然语言。

1．词袋模型（Bag Of Words，BOW）

词袋模型最初被用于自然语言处理和信息检索（Information Retrieval，IR）领域，是信息检索领域常用的文档表示方法。词袋模型基于文本中每个词的出现都不依赖于其他词是否出现的假设，在进行文档表示时，忽略文本的词序、语法和句法，而将文本看作由词组成的集合。也就是说，词袋模型认为文档中任意位置出现的任何单词，都与该文档的语义无关。例如，有如下两个文档：

1：北京今天下雨，深圳也下。

2：北京和深圳今天都下雨。

我们基于这两个文本文档，构造一个词典：

Dictionary = {1.“北京”, 2. “今天”, 3. “下”, 4. “雨”, 5. “深圳”, 6. “也”, 7. “和”,8.“都”}，其中数字表示每个汉字的索引，例如北京是第 1 个单词。

这个词典包含 8 个不同的单词，利用词典中单词的索引号，可以用 8 维向量表示上面两个文档，向量中的整数数字表示对应索引号的单词在文档中出现的次数：

1：[1, 1, 2, 1, 1, 1, 0, 0]

2：[1, 1, 1, 1 ,1, 0, 1, 1]

例如，“下”这个词在第一句话中出现了两次，因此在向量的第 3 个位置的数字是 2，而“都”没有出现，因此在向量的第 8 个位置的数字是 0。

通过上述例子可以知道，通过词袋模型，一个文档可以转化为一个向量，向量中的每个元素表示词典中相应元素在文档中出现的次数，这种处理方式使得我们可以较方便地将源文档模型化。同时，构造词袋模型文档表示向量的过程中我们也可以明显感觉到，词袋模型并没有表达单词在原来句子中出现的次序，这是词袋模型的缺点之一。例如，在新闻个性化推荐中，假设用户对“美国双子塔爆炸事故”感兴趣，那么使用忽略词汇顺序和语法的词袋模型表示这个短语，会导致系统认为用户对“美国”“爆炸”“事故”“双子塔”感兴趣，进而向用户推荐“美国交通爆炸事故”等内容，这个推荐明显不合理，因为用户真正感兴趣的可能是“美国袭击爆炸事故”。简言之，BOW 模型是否适用，需要根据实际情况确定。那些不可以忽视词序、语法和句法的场合均不能采用 BOW 的方法。

2．TF-IDF

TF-IDF（Term Frequency-Inverse Document Frequency，词频−逆文档频率）是一种基于统计的加权方法，常用于信息检索领域，用具体词汇在文档中出现的次数和该词汇在语料库中出现的次数两个值评估该词汇对相关文档的重要程度。TF-IDF 常被搜索引擎用来评估文档与用户查询之间的相关程度。对于指定

的文档，TF（Term Frequency，词频）指某词语在该文档中出现的次数，IDF（Inverse Document Frequency，逆文档频率）是词语普遍重要性的度量。词汇在指定文档内的高 TF，以及该词汇在整个文档集合中的高 IDF，将使该词汇在该文档内有较高权重的 TF-IDF。TF-IDF 倾向于过滤常见词汇、保留重要词汇的做法是由其主要思想决定的，TF-IDF 的核心思想是：在一篇文档中出现频率高且在其他文档中很少出现的词汇有较好的类别区分能力，适合用于文档分类。

实际上，在同一类文档中频繁出现的词汇，往往具备代表该类文档的特征的作用，这样的词汇应具有较高的权重，并作为该类文档的特征词。这也是 IDF 的不足之处。

3．词嵌入

用词嵌入（Word Embedding）表示单词是将深度学习引入自然语言处理的核心技术之一，而词嵌入来源于一个非常朴素的思想：欲在自然语言理解领域使用机器学习技术，则需要找到一种合适的、将自然语言数学化的方法。研究人员最初使用独热表示（one hot representation）方法，即使用词表大小维度的向量描述单词，每个向量中多数元素为 0，只有该词汇在词表中对应位置的维度为 1。假设有一个词表 H，H 包含 *N* 个词汇，词汇“雨伞”是词表 H 中的第 2 个词汇，词汇“伞”是词表 H 中的第 4 个词汇，则

词汇“雨伞”的独热表示表示为：[0 1 0 0 0 0...]

词汇“伞”的独热表示表示为：[0 0 0 1 0 0...]

同时，可以给词表中的每个词汇分配 ID，“雨伞”的 ID 为 2，“伞”的 ID 为 4。

独热表示法将所有词汇单独考虑，仅从词汇的上述向量表示难以发现词汇间的同义、反义等关系。另外，由于上述词汇的表示方式过于稀疏，在具体处

理时极易造成维度灾难。词嵌入法在基于独热表示法的基本思想的同时，增加了单词间的语义联系，并降低了词向量维度以避免维度灾难。

通过训练获得词向量的方法有很多，接下来将简单介绍通过训练获得词向量的主要思想。

一个包含 t 个词汇 $\omega_1,\omega_2,...,\omega_t$ 的句子是自然语言的概率可以表示为

$$\begin{aligned} & p(\omega_1,\omega_2,...,\omega_t) \\ &= p(\omega_1)\times p(\omega_2\mid\omega_1)\times p(\omega_3\mid\omega_1,\omega_2)\times...\times p(\omega_t\mid\omega_1,\omega_2,...,\omega_{t-1}) \\ &\cong p(\omega_t\mid\omega_1,\omega_2,...,\omega_{t-1}) \end{aligned}$$

上述表示也被称为语言模型。对于 N-gram 模型来说，

$$p(\omega_1,\omega_2,...,\omega_t)\cong p(\omega_t\mid\omega_{t-n+1},\omega_{t-n+2},...,\omega_{t-1})$$

加拿大蒙特利尔大学教授 Yoshua Bengio 发表了使用三层神经网络构建语言模型的研究。该研究中的第一层是输入层，输入句子中已知的前 n–1 个词汇的词向量，且将这 n–1 个词向量拼接成 1 个向量；第二层是神经网络模型的隐藏层；第三层是输出层，其中第 i 个节点的值等于下一个词为 ω_i 的概率的对数。在对模型进行优化的过程中，同时对单词的词向量进行优化。当模型优化结束时，即可获得语言模型和词向量。

基于深度学习的自然语言理解是较新的研究方向。获得自然语言的向量化表示后，通过采用端到端的方法直接生成回复，其最典型的框架是 Encoder-Decoder。Encoder-Decoder 框架是文本处理领域的一种研究模式，其不仅可以应用在聊天机器人领域，还可以应用在机器翻译、文本摘要、句法分析等应用场景中。我们可以将 Encoder-Decoder 框架理解为一个适用于处理由句子（或篇章）X 生成句子（或篇章）Y 的通用模型。对于句子对 (X,Y)，我们的目标是给定输入句子 X，通过 Encoder-Decoder 框架生成目标句子 Y。X 和 Y 可以是同一种语言，也可以是两种不同的语言。当 X 和 Y 是不同的语言时，可以将

Encoder-Decoder 框架理解为一个自动翻译器。构成 X 和 Y 的单词序列可以分别表示为

$$X =< x_1, x_2, ..., x_m >$$
$$Y =< y_1, y_2, ..., y_n >$$

编码器 Encoder 负责对输入句子 X 进行编码，通过非线性变换将输入句子转化为中间语义表示 C：

$$C = \Re(x_1, x_2, ..., x_m)$$

解码器 Decoder 负责根据 Encoder 生成的中间语义表示 C 和之前已经生成的历史信息 $y_1, y_2, ..., y_{i-1}$，生成 i 时刻的单词 y_i：

$$y_i = \Im(C, y_1, y_2, ..., y_{i-1})$$

图 2-5 描述了抽象化的 Encoder-Decoder 框架。

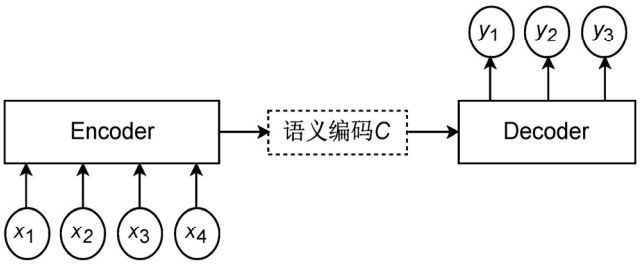

图 2-5　抽象化的 Encoder-Decoder 框架

依次产生 y_i 的过程，看起来就是整个系统根据输入句子 X 生成目标句子 Y 的过程。

在词嵌入技术得到发展的同时，语言模型的研究也有了很大进展。2018 年 10 月，谷歌提出了通用的语言模型 BERT[1]，BERT 模型不仅能解决 11 种不同的自然语言处理任务，而且在所有任务上的精度均大幅领先于其他模型，甚至

在某些方面超越了人类。

不管是基于检索的聊天机器人还是基于生成的聊天机器人，都可以在研发过程中使用上述 Encoder-Decoder 框架技术。

对基于生成的聊天机器人来说，在使用上述 Encoder-Decoder 框架解决核心技术问题时，需要注意多轮对话、安全回答和个性一致等问题，这些需要注意的问题也是目前基于深度学习的端到端聊天机器人的热门研究方向。

1．多轮对话问题

基于 Encoder-Decoder 框架，聊天机器人作为一个有效的对话系统，可以根据用户当前的输入信息自动生成应答回复。但是，一般情况下，人们聊天并不是单纯的一问一答，回答的内容通常要参考上下文信息，即在用户输入当前问句信息之前聊天机器人和用户的对话信息。由于这个过程中存在多轮的一问一答，这种情况一般被称为**多轮对话**。

利用深度学习技术解决多轮对话问题的关键是将上下文聊天信息引入 Encoder-Decoder 模型中。按照之前的讲述，上下文聊天信息是除了当前输入信息的额外信息，有助于 Decoder 生成更好的应答回复，因此上下文聊天信息应该被引入 Encoder 编码器中。解决多轮对话问题的一种思路是：将上下文聊天信息和本轮用户输入的信息拼接到一起，形成一个更长的输入提供给 Encoder，就可以把上下文信息融入 Encoder-Decoder 模型中。对基于 RNN（Recurrent Neural Network，递归神经网络）模型的 Encoder 来说，采用上述方式使得 RNN 模型的输入非常长。众所周知，RNN 模型的效果随着输入线型序列长度增长而降低。所以，简单地拼接上下文聊天信息和本轮用户输入信息的策略无法产生理想的聊天效果。

图 2-6 给出了一个同时考虑词语级别和句子级别的多轮对话模型[2]。这项

工作由微软小冰团队提出，先对不同级别的词语和句子进行匹配，然后按照时间序列整合匹配结果。

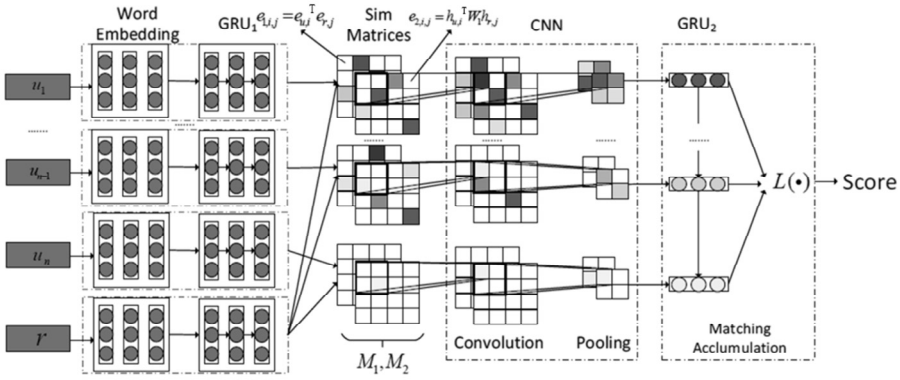

图 2-6　同时考虑词语级别和句子级别的多轮对话模型

考虑到 RNN 模型对过长输入敏感的问题，许多研究者提出了针对聊天机器人场景优化的 Encoder-Decoder 模型，核心思想是用多层前向神经网络代替 RNN 模型。多层前向神经网络的输出代表上下文聊天信息的内容和当前输入信息的中间语义表示，而 Decoder 依据这个中间表示生成对话回复。这样做既能将上下文聊天信息和当前输入语句信息通过多层前向神经网络编码成 Encoder-Decoder 模型的中间语义表达，又避免了 RNN 模型对过长输入敏感的问题。

解决多轮会话中上下文问题的另外一种思路被称作**层级神经网络**（Hierarchical Neural Network，HNN）。HNN 方法在本质上也可被看作 Encoder-Decoder 框架。HNN 的主要特征是其 Encoder 采用了二级结构：使用第一级句子 RNN（Sentence RNN）对句子中的每个单词进行编码，形成每个句子的中间表示（可以将这里的中间表示理解为前述 Encoder-Decoder 模型中的 Encoder 生成的中间语义表示 *C*），第二级句子 RNN 则按照上下文中句子出现的先后顺序序列对第一级句子 RNN 的中间表示进行编码，这级 RNN 模型可称作上下文 RNN（Context RNN），其尾节点处隐层节点状态信息就是所有上下

文聊天信息及当前输入信息的语义编码，作为 Decoder 层的输入之一，生成单词序列，这样就可以在生成应答回复时将上下文信息考虑进来。例如，参考文献[3]采用基于层次的 RNN，通过语料分析及推理机制和行为生成机制学习得到状态和动作的空间表示，并将多轮对话信息编码成一个稠密的空间向量，映射到对话的上下文，用于对下一轮对话语言中的片段进行解码。该模型中的编码 RNN 对出现在多轮对话中的语言片段进行编码，上下文 RNN 对其中的时间进行编码，解码 RNN 负责对下一时刻的对话回复进行预测。根据经验我们知道，相比直接将上下文信息和本轮对话信息串联拼接，层次 RNN 对信息的传递是有损的，因此在实际工程中应用该模型前，需要根据具体的项目要求对模型进行适配。参考文献[3]的模型表示如图 2-7 所示。

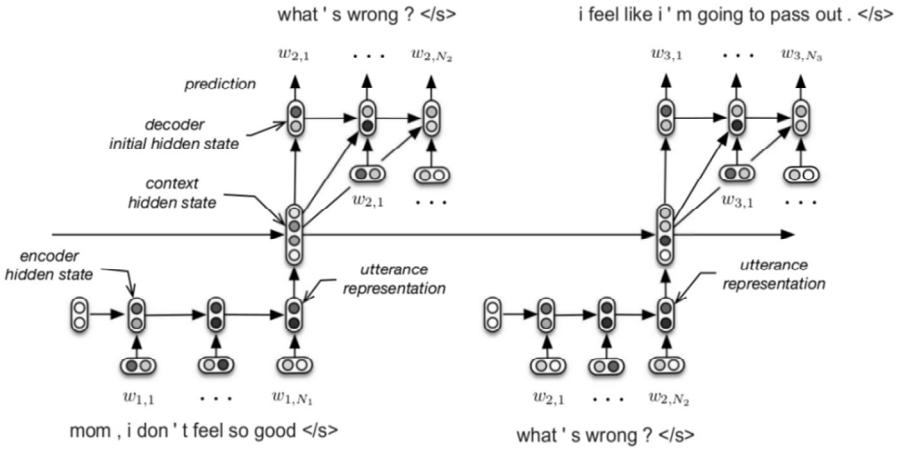

图 2-7　参考文献[3]的模型表示

综上所述，利用深度学习模型解决多轮对话的上下文信息问题时的一个典型做法是：在 Encoder 阶段将上下文聊天信息及当前输入信息同时编码，参考上下文信息生成应答回复。

2．避免安全回答

采用经典的 Encoder-Decoder 模型构建开放领域生成式聊天机器人系统，

比较容易产生的另一个问题是“安全回答”。“安全回答”问题是指聊天机器人的大多数答案严重趋同，且不具有实际价值，无法让人机对话继续进行。也就是说，不论用户输入什么内容，聊天机器人都用少数常见的句子进行应答，比如英文的“I don’t know”“Sure”、中文的“是吗”“呵呵”“嗯”等。虽然在很多情况下，聊天机器人这样的回答也不能说是错误的，但用户不会满足于这种程度的聊天。这个现象产生的主要原因是 Encoder-Decoder 模型训练时使用的聊天训练数据中确实包含很多宽泛而无意义的应答，聊天机器人通过 Encoder- Decoder 模型学会了这种常见应答模式。如何解决“安全回答”问题，让机器产生多样化且有意义的应答是聊天机器人领域的一个重要的研究课题。“安全回答”问题是多种因素综合作用的结果，其中的症结是：训练数据中的词语在句子不同位置的概率分布体现出明显的长尾特性，词语概率分布上的模式会优先被 Decoder 学到，并在生成过程中抑制“问题”和“回复”之间词语关联模式的作用。从全局的角度考虑“安全回答”问题，其根本上是神经网络回复生成模型陷入了局部最优解，可以通过给模型施加一个干扰使其跳出局部解。基于这个思想，研究人员将生成对抗网络（Generative Adversarial Networks，GAN）引入聊天回复生成系统以解决安全回答的问题。

下面介绍李纪为博士（斯坦福大学博士，毕业后创办了香农科技）于 2017 年发布的研究[4]，这项研究的主要贡献是使用对抗训练（Adversarial Training）的思路解决开放领域的对话生成（Open-domain Dialogue Generation）问题。其主要思想是将整体任务划分到生成器（Generator）和判别器（Discriminator）两个子系统上，生成器利用 seq2seq 模型以上文的句子作为输入，输出对应的对话语句；判别器用来区分在前文条件下生成的问答是否与人类行为接近。两者结合的工作机理也很直观，生成器不断根据前文生成答句，判别器则不断用生成器的生成作为负例，原文的标准回答作为正例来强化分类器。在训练的过程中，生成器不断改良答案来欺骗判别器，判别器不断提高自身的判别能力，直至两者收敛达到某种均衡。以这样一种博弈式的训练方式来优化无监督的开

放领域的人机对话任务显然是很有意义的。不过，由于 GAN 和自然语言处理很难很好地融合，作者采用了强化学习的方法来规避 GAN 在自然语言处理中使用的难点。

3．个性一致问题

对具有情感陪伴、虚拟个人助理等功能的聊天机器人应用来说，聊天机器人往往会被用户当作一个具有个性的虚拟人，比如用户会问聊天机器人“你的生日是什么时候”“你的爱好是什么”“你的家乡在哪里”“你多大”等问题。如果将聊天机器人看作一个虚拟人，那么这位虚拟人的年龄、性别、爱好、习惯、语言风格等个性特征信息应该具有一致性。seq2seq 模型训练的都是单句信息对单句回复的映射关系，并没有统一维护聊天机器人个性信息的功能，无法保证对用户的相同语义的问题每次都能产生完全一致的应答，因此利用经典的 seq2seq 模型训练出的聊天机器人很难保持个性信息的一致。另外，在具体定义聊天机器人时，往往面对不同用户喜欢不同的聊天风格或者不同身份的聊天机器人的情况。

那么，如何在 seq2seq 框架下维护聊天机器人个性信息的一致性呢？一种比较直观的解决方案是，在聊天机器人系统中定义聊天机器人的个性化信息，这些预定义的个性化信息通过词嵌入表达方式体现。在这种情况下，聊天机器人的整体技术框架仍然采用 seq2seq，其实现思路是把聊天机器人的个性信息导入 Decoder。也就是说，在采用基于 RNN 模型的 Decoder 生成回答的时候，每个 t 时刻，神经网络节点除了接受 RNN 标准的输入，也将预定义的个性化词嵌入信息作为输入。通过这种方式，可以引导聊天机器人系统在输出回复时倾向于输出符合聊天机器人身份特征的个性化信息。

根据上述思路，还可以衍生出很多种其他深度学习框架下维护聊天机器人个性一致的技术框架方案。这些技术框架方案的核心思想是把聊天机器人的个

性信息在 Decoder 阶段体现出来，以达到维护个性一致的目的。如参考文献[5]通过建立基于个性化的对话模型，尝试为聊天机器人进行个性化人格建模。

2.1.4 基于知识图谱的自然语言理解

要讲述基于知识图谱的自然语言理解，就需要先对知识图谱的知识表示、知识构建和知识融合进行阐述。

知识图谱作为实现智能化语义检索的基础和桥梁，由谷歌于 2012 年提出。无论是在聊天机器人领域还是在其他应用领域，知识图谱的重要性不言而喻。一方面，知识一直以来都是人工智能研究的中心课题，知识图谱作为知识的载体必然成为研究的重点和难点。另一方面，知识作为智能系统的强力助推器，可以很好地辅助互联网应用，支持国家“互联网+”战略，产生更大的社会效益和经济效益。

知识图谱可以被看作结构化的语义知识库，旨在以符号形式描述真实世界中存在的各种实体或概念及其相互关系，其基本组成单位是“实体—关系—实体”形式的三元组，以及实体及其相关属性的“属性—值”对（Attribute-Value Pair，AVP）。知识图谱中的每个实体或概念可以用一个全局唯一确定的标识符来标识；每个“属性—值”对都是对实体内在具体特性的刻画；使用关系连接两个实体，刻画实体之间的关联，构成网状的知识结构。从本质上讲，知识图谱可以被看作一张巨大的、包含节点与边的图，其中的节点表示物理世界的实体或概念，而网络中的边代表了实体间的各种语义关系，这个图模型可用 W3C 提出的资源描述框架（Resource Description Framework，RDF）或属性图表示。

知识图谱的建立涉及一系列结构化和非结构化数据的处理，具体包括知识的表示、提取、存储、检索等技术。知识图谱是知识表示与推理、数据库、信

息检索、自然语言处理等多种技术发展与融合的产物。在实际工程中应用知识图谱时，特别是在互联网应用和聊天机器人的应用场景下，需要克服传统知识处理方法实施成本高、技术周期长、人才缺乏、基础数据不足等制约，充分利用成熟的工业技术，从实际的业务出发，循序渐进地推进知识图谱相关工程的实施。

在互联网飞速发展的今天，知识大量存在于非结构化的文本数据、半结构化的表格和网页及生产系统的结构化数据中。构建知识图谱不仅需要结合文本、多媒体、半结构化或结构化知识、服务（或 API）、时态知识等多种形式的知识，对这些知识进行统一的知识表示，还需要在此基础上结合结构化（如关系型数据库）、半结构化（HTML 或 XML）和非结构化（文本、图像）等多源异质数据源构建专业领域及开放领域的知识库等。知识图谱的主要模块如图 2-8 所示。

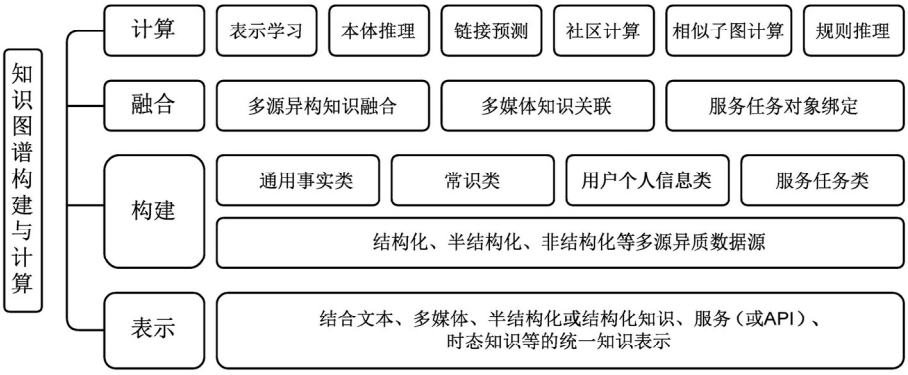

图 2-8　知识图谱的主要模块

在构建知识图谱时，要着重考虑通用事实类、常识类、用户个人信息类、服务任务类等不同类型的知识库。显然，不同类型的数据和不同种类的知识适用的构建技术有所不同，如可以使用知识映射的方式处理结构化数据、使用包装器（wrapper）处理半结构化知识、使用文本挖掘和自然语言处理技术处理非结构化知识。例如，处理非结构化数据时，需要在提取其正文后通过实体识别

技术提取正文中的实体。常用的实体识别方法有两种：使用实体链接将文章中可能的候选实体链接到预置的知识库上；当没有预置知识库时，要先使用命名实体识别技术识别文章中的实体，再通过实体关系识别、事件抽取等技术获得知识。

1．知识图谱介绍之知识表示

知识表示即知识在计算机内的存储和处理格式。知识图谱是图形式的知识库，其以关系连接头尾实体构成有向图。一般来说，我们使用三元组表示一条知识，头尾实体是图谱的节点，关系是图谱的边。

知识表示使用的数据结构，最常见的是图（graph）和树（tree）。知识表示的图形式中包含“有类型的边”，其中的每个边和每个节点都拥有元数据。用图形式表示知识，丰富的知识结构主要表现为图上的边，此时各种推理算法的作用是在图上推导出新的边。现有的图形式数据库的缺点主要表现为：在知识表示方面存在一定局限，导致工程上实施时需要投入的成本很高；难以对结构化数据和非结构化数据进行混合表示，这个问题影响了图对知识的表示能力。

在现实的工程中，为了克服知识提取所需成本较高的问题，往往不会追求一步到位生成纯结构化的数据表示，知识库中的数据往往由结构化和非结构化（主要是文本）两种类型的数据混合而成。实际上，结构化数据和非结构化数据的混合表示，使知识库的图形式较为复杂，因此工程上最广为接受的知识表示是树结构，其中树形的 JSON 满足了结构化和非结构化混合表示的需要，是目前工程中常用的知识表示方式。这种方式存在的主要缺陷是：不能很好地与机器学习技术结合，阻碍了知识图谱更广泛的应用；逻辑推理时间复杂度高，在当前的大数据实际场景中难以得到应用。

针对上述问题，学术界提出了基于几何空间的知识表示方法。在这种表示方法中，每一个实体都可以看作几何空间中的一个点，每一个关系都可以看作集合空间中的一个平移向量，每一个元组都以平移原则作为基本几何表示形式：

头实体可以按照关系向量移动到尾实体。基于这种方法，不仅可以设计出高效的、基于统计学习的人工智能算法，还可以提高知识库的泛化性以解决实际工程中出现的知识图谱补全等问题。

2. 知识图谱介绍之知识构建

基于已融合数据构建知识库的具体方法，根据已融合数据结构的不同而变化。对于已存在于传统数据库中的结构化数据，使用简单的映射，按照特定的结构将其中的知识映射到知识图谱中即可。对于 HTML 等半结构化数据，可以通过包装器结合半结构化数据的模式信息，通过定义或者人工智能的方法获得抽取规则，然后将按规则抽取的信息存储到特定格式的知识图谱中。对于非结构化数据，可以通过文本挖掘技术发现文本隐含的模式，例如通过自然语言处理方法对文本数据中的开放领域知识进行抽取、通过使用模式识别和数字图像处理技术对图像数据进行处理。

总而言之，知识的构建方法，随着融合后数据的结构和数据类型的不同而不同。

3. 知识图谱介绍之知识融合

构建知识图谱的过程中，在从各个数据源获取知识后，需要将这些知识融合到一起，形成一个统一的知识库。将从多个数据源抽取的知识进行融合的过程就叫作**知识融合**。**本体**（ontology）是知识融合过程中的重要概念。本体不仅提供了统一的术语字典，还构建了各个术语间的关系及限制，提供了根据具体业务建立或者修改数据模型的功能。通过映射建立本体中的术语与从不同数据源中抽取的知识所包含的实体之间的对应关系，完成不同数据源知识的融合。不同数据源中指向现实世界同一客体的实体可能出现描述不同或名称不同的情况，因此需要通过实体匹配和本体融合技术将不同数据源中描述相同实体的知识进行融合。在知识融合技术中，本体匹配为概念或者实体之间的对应提供了基

础。已有的本体匹配算法大致可分为模式匹配（schema matching）和实例匹配（instance matching）两种，也有少量研究致力于将模式和实例的匹配相融合。下面围绕模式匹配和实例匹配介绍具有代表性的几项研究。

模式匹配的主要任务是寻找本体中属性和概念之间的对应关系[6]。大规模的本体匹配一般使用锚（anchor）[7]相关的技术，将来自两个本体的相似概念作为起点，根据这两个相似概念的父概念、子概念等邻居信息逐渐构建小的相似片段，进而从中找出匹配的概念。同时，利用迭代的思想，将新找出的匹配的概念对作为新的锚，再根据新锚相关本体的邻居信息构建新的片段，这个迭代的过程不断重复，直到找不到新的匹配概念为止。采用分而治之的思想[8]处理大规模本体匹配的问题也是本体匹配领域常用的方法。在具体操作时，先根据本体的结构对其进行划分，获得组块，再对从不同本体获得的组块进行基于锚的迭代匹配，最后从匹配的组块中寻找对应的概念和属性。

实体匹配可以评估来自不同异构数据的实例对的相似度，评估的结果被用来判断这些实例是否指向给定领域的相同实体。利用局部敏感哈希（Locality-Sensitive Hashing）技术提高实例匹配的可扩展性[9]的方法，与使用向量空间模型表示实例并基于规则采用倒排索引（inverted index）获取最初的匹配候选[10]的方法都是实体匹配领域的典型技术。

虽然研究人员已经公布了多项可处理大规模本体的实体例匹配算法，但是同时保证效率和精度仍是实体匹配领域中具有挑战性的目标。Shao 等人[11]于2016 年提出了一种可迭代框架技术，该技术充分利用特征明显的已有匹配方法来提高本体匹配的效率。同时，基于相似度传播的方法，利用自定义的加权指数函数来提高实体匹配的精度。

为了得到尽可能完善的融合知识库，除了需要融合离线的多源异构知识，还要考虑在线的服务和任务类动态知识对象的绑定。这部分工作相当于根据具体

的交互需要，在线动态扩充知识图谱并对知识图谱进行实例化的过程。

了解了上述知识图谱的相关知识后，我们需要思考的问题是，聊天机器人对知识图谱有哪些特殊的需求？

1．聊天机器人需要更个性化的知识图谱

对聊天机器人来说，除了需要实体知识图谱和兴趣知识图谱等开放领域稀疏大图，还需要针对机器人和用户的个性化稠密小图。机器人或 Agent 需要相应的知识图谱来建模并展示自我认知能力，而用户知识图谱则可以被看作更精细化和个性化的用户画像知识表现。图 2-9 对比了开放领域稀疏大图和描述机器人属性及用户属性的个性化稠密小图的知识图谱。

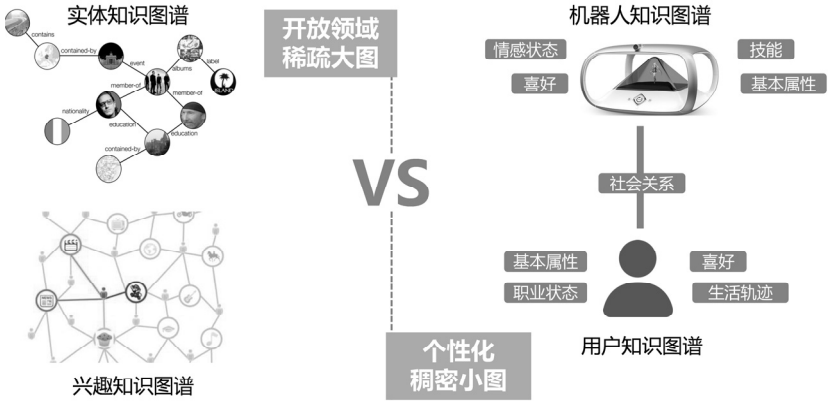

图 2-9　开放领域稀疏大图与个性化稠密小图的对比

机器人“琥珀·虚颜”，有自身的情感状态、喜好、技能等知识维度，而用户则需要表达其职业状态和生活轨迹等信息。需要强调的是，无论是个性化小图还是开放领域大图，都不是独立存在的，在实际工程应用时需要将它们融合在一起才能发挥更大的价值。例如，机器人喜欢的明星需要和实体知识图谱中的明星娱乐图谱关联；同样，机器人的爱好需要与兴趣图谱关联；机器人需要与用户形成亲人、好友、雇佣等社会关系。

2．聊天机器人不仅需要静态知识图谱，还需要动态知识图谱

若一个聊天机器人想要更像人，就需要从早到晚做不同的事情，也就是需要有自己的生活规律，研发时又该如何刻画这个聊天机器人的生活轨迹呢？例如，图 2-10 所示的聊天机器人自身的生活规律和用户的实时状态，应该如何刻画?

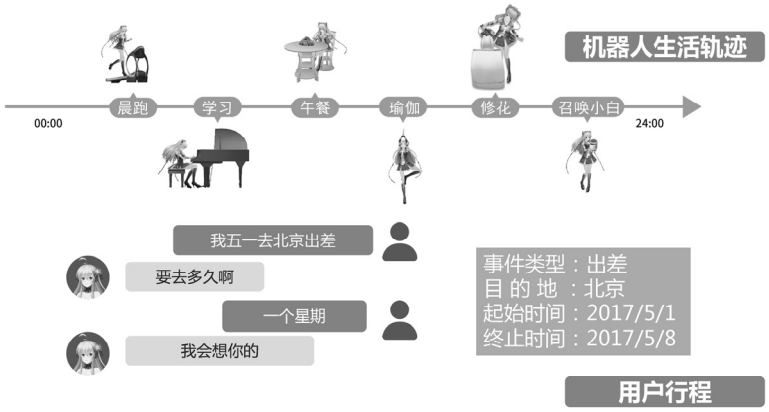

图 2-10　聊天机器人自身的生活规律和用户的实时状态

显然，想让聊天机器人有生活规律，就需要在研发过程中在图谱中体现时态知识。另外，聊天机器人作为用户的个人助理，需要记住用户图谱——各种用户已经发生、正在发生或即将发生的事件。知识图谱中的用户行程不仅是一个关系或属性，还是一个由多元（N-ary）数据组成的事件。为了表示用户行程，研发过程中需要定义多种事件类型，并在时间和空间两个维度上对用户的各种动态进行刻画。

3．聊天机器人不仅需要表达客观知识的知识图谱，还需要可以刻画主观情感的知识图谱

聊天机器人不能只是冷冰冰地回答用户的问题或帮助用户完成特定功能，它需要感知用户的情感并在输出答案回复的同时体现出相应的情感，这样拟人化程度才更高。图 2-11 所示为聊天机器人在与用户交互的过程中，根据感知到的用户情感状态，与用户进行交互的示例。

已有的知识图谱大多是客观的，即用于描述一些客观的事实。如何使聊天机器人结合个性化图谱，尽量形成一些主观认知，进而刻画机器人或用户的情感元素呢？例如，用户说“我心情不好”这属于闲聊中的情感表达范畴，需要机器人将用户当前的心情状态更新到用户知识图谱的对应维度数值中。相应地，机器人也会有自己的心情、体力，甚至和用户之间的好感度关联。当机器人心情不错，同时和用户很亲密时，它就会主动关心用户。结合机器人和用户情感因素的动态回复会更加温馨且贴合场景。另外，在多轮对话时，当用户说“来一首快乐的歌吧”时，需要进一步结合音乐知识的知识图谱（快乐作为歌曲的曲风或风格标签）和用户知识图谱中的音乐偏好，推荐符合用户喜好的歌。

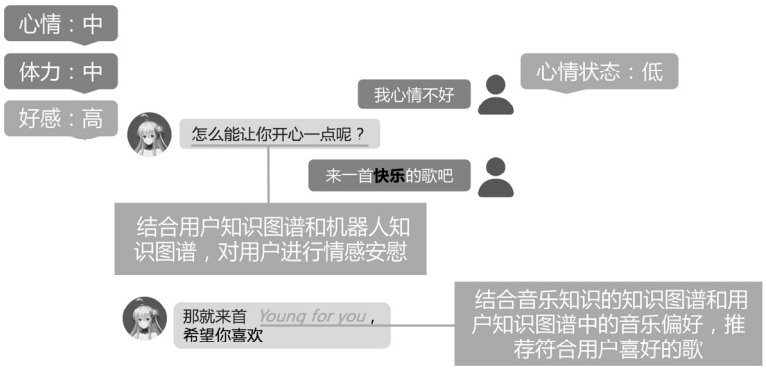

图 2-11　聊天机器人与用户进行交互的示例

4．聊天机器人为了完成用户要求需要对接外部服务或开放 API

此时，要将传统的关系型知识图谱（刻画二元关系）扩展到支持动态服务的动态图谱（刻画多元关系，事件属于服务图谱的一个特例）。同时，如何刻画服务之间的各种关系（如因果、时序依赖等）也是图谱扩展过程中需要考虑的。例如，完成订餐后，会有很多后续（follow-up）服务（订花或预约车等）可供消费。建立这些服务之间的关联对进行精准的多轮对话过程中的场景切换是非常必要的，图 2-12 所示为一个案例。

图 2-12　聊天机器人需要对接外部服务或开放 API

5．聊天机器人不仅需要纯文本的知识图谱，还需要包含多媒体知识的知识图谱

人类不仅把文字作为接触世界的手段，还会结合图像、语音和文字等多模态来了解外部世界。因此，在研发聊天机器人时所构建的知识图谱也应该从单纯文本自然扩展到多媒体知识图谱。斯坦福大学李飞飞教授创办的 ImageNet 和 Visual Genome 正是在这方面进行努力的典范。对于用户图谱这样更新频度非常高且很稠密的知识图谱，多媒体知识的引入能帮助聊天机器人从更多的维度了解用户，并提供诸如 Visual QA 等潜在的问答服务。例如，小明正在和聊天机器人进行交互，聊天机器人通过自身搭载的摄像头识别出当前交互的用户是小明，然后根据小明的图像与用户 ID 的关联，进一步得到自身保存的与小明相关的长短时记忆，了解到他将在 4 月 20 日～23 日去南京出差，而 4 月 24 日要和小兰共进晚餐。此时，通过用户知识图谱中的社交关系了解到小兰是小明的女友。当聊天机器人需要进一步了解小兰长什么样时，或者当小兰出现在聊天机器人面前时，聊天机器人需要认出小兰，这时就需要用到包含多媒体知识的知识图谱。图 2-13 是对上述描述的可视化说明。

总而言之，聊天机器人需要基于多源、异构的数据构建包含多类别且体现动态和个性化的知识图谱。这其中包括基于来自互联网的数据刻画世界知识、

基于用户数据刻画用户画像知识、针对机器人的各种基本属性、社会关系、情感状态、兴趣爱好、日常生活等静态和动态知识得到的融合图谱，它是时空坐标中针对特定交互场景和时间节点的一个镜像，图 2-14 所示为一个案例。

图 2-13　聊天机器人需要包含多媒体知识的知识图谱

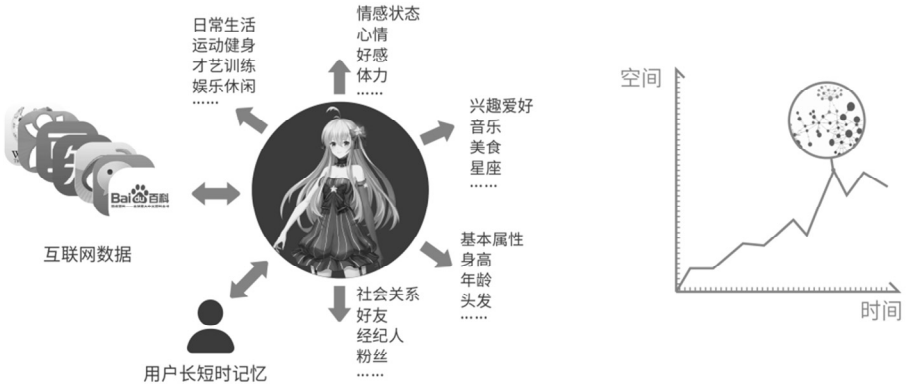

图 2-14　聊天机器人需要基于多源、异构的数据构建包含多类别且体现动态和个性化的知识图谱

根据聊天机器人所属领域的自然语言理解具体技术的需要，建立知识图谱后，使用分词、实体识别与消歧等技术，将用户输入的自然语言中包含的实体与知识图谱中的实体进行链接，使机器人可以理解用户输入的自然语言中包含的意图，从而从知识图谱中抽取合适的内容对用户输入进行回复。

2.2 自然语言生成

2.2.1 自然语言生成综述

自然语言生成作为人工智能和计算语言学的分支，其对应的语言生成系统可以被看作基于语言信息处理的计算机模型，该模型从抽象的概念层次开始，通过选择并执行一定的语法和语义规则生成自然语言文本。自然语言生成系统的主要架构可以分为流线型（pipeline）和一体化型（integrated）两种，流线型的自然语言生成系统由几个不同的模块组成，每个模块之间的交互仅限于输入输出，各模块之间不透明、相互独立；而一体化型的自然语言生成系统的模块之间是相互作用的，当一个模块内部无法做出决策时，后续模块可以参与该模块的决策。一体化型的自然语言生成系统更符合人脑的思维过程，但是实现较为困难，现实中较常用的是流线型的自然语言生成系统。流线型的自然语言生成系统包括文本规划、句子规划、句法实现 3 个模块。文本规划决定说什么，句法实现决定怎么说，句子规划则负责让句子更加连贯，其具体架构如图 2-15 所示。

自然语言生成和自然语言理解都是自然语言处理的分支，且从表面上看自然语言生成是自然语言理解的逆过程，但实际上二者的侧重点不同，自然语言理解实际上是使被分析的文本的结构和语义逐步清晰的过程，而自然语言生成的研究重点是确定哪些内容是满足用户需要必须生成的，哪些内容是冗余的。尽管研究的侧重点不同，但自然语言生成与自然语言理解有诸多共同点：其一，二者都需要利用词典；其二，二者都需要利用语法规则；其三，二者都要解决指代、省略等语用问题。

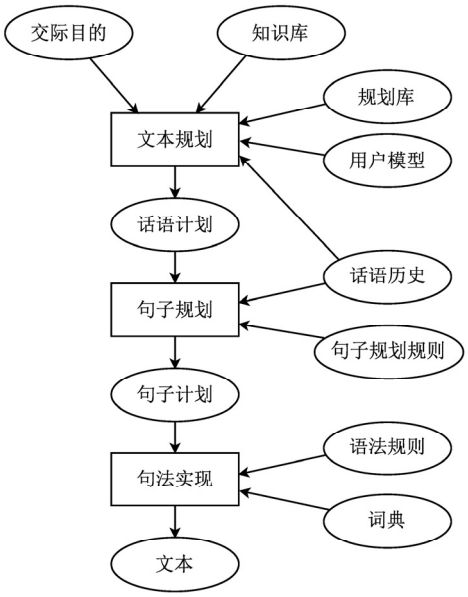

图 2-15　流线型的自然语言生成系统架构图

虽然自然语言生成是聊天机器人系统中的一个重要模块，但目前对自然语言生成的研究进展远不如自然语言理解。目前，多数聊天机器人系统使用的对话生成技术主要包括检索式和生成式两种。

1．检索式对话生成技术

检索式对话生成的代表性技术是通过排序技术和深度匹配技术在已有的对话语料库中找到适合当前输入的最佳回复。这种方法的局限性体现在仅能以固定的语言模式对用户输入进行回复，而无法实现词语的多样性组合，因此无法满足回复的多样性要求。

2．生成式对话生成技术

生成式生成的代表性技术是从已有的“人—人”对话中学习语言的组合模式，是在一种类似机器翻译中常用的“编码—解码”的过程中逐字或逐词地生成回复，生成的回复有可能是从未在语料库中出现的、由聊天机器人自己“创

造”的句子，由此使机器人具备了造句的能力。需要注意的是，在使用生成式对话生成技术生成答案的过程中，机器并未像人类一样试图理解句子的语法和词汇的词性。

自然语言生成还面临如下挑战。

（1）涉及文法开发，需要将文法结构和应用特有的语义表征相关联，但由于自然语言中存在海量的文法结构，造成搜索空间巨大，如何避免生成有歧义输出成了一个有挑战的问题。

（2）由于语言的上下文敏感性，生成语言时如何整合包括时间、地点、位置、用户信息等在内的上下文信息也是一个难题。

（3）基于深度学习技术生成回复的对话模型很难解释，也很难被人类理解，只能通过更好的语料和参数调整来改善对话模型。

2.2.2 基于检索的自然语言生成

基于检索的自然语言生成并不是如字面意思一样生成自然语言，更多是在已有的对话语料库中检索出合适的回复。这种方法只能以固定的语言模式对用户的输入进行回复，其表现依赖于已有对话库。当对话库不能覆盖可能的对话场景时，用户体验将会受到影响。

虽然基于检索的生成方式存在依赖于对话库、回复不够灵活等缺陷，但由于其实现相对简单、容易部署，在实际工程中得到了大量的应用。例如，银行在线客服系统问答场景中使用的基于检索的自然语言生成系统，会根据用户的输入，在已有问答库中检索相关的回复，当检索不到合适的回复时，这类系统一般会选择将用户的问题转发至人工在线客服。

2.2.3　基于模板的自然语言生成

基于模板的自然语言生成技术中使用的模板往往由语言学家参与整理，在定义这些模板的结构时，应当力求让语言学家容易了解和书写这些模板。但是一般的模板系统大多从实现的角度进行描述，造成了模板描述语言对于用户（语言学家）而言不够自然的问题，影响了该类系统的可维护性和可扩充性。

所以，用于自然语言生成的模板描述语言首先应当符合人的思维习惯，使得模板易于书写。其次，用于自然语言生成的模板描述语言应当有较强的描述能力，能够表达尽可能多的语言现象。另外，实际应用中需要模板描述语言易于扩充，以便在其中加入新的成分，使其能够描述现有语言成分无法描述的新的语言现象，保证模板描述语言的健壮性。

自然语言生成模板由句子（sentence）模板和词汇（word）模板组成。句子模板包括若干个含有变量的句子，词汇模板则是句子模板中的变量对应的所有可能的值。为了方便理解，图 2-16 和图 2-17 分别提供了询问天气场景中的句子模板和词汇模板的可视化表示。

```
Topic->weather
  Act->query
    Content：weather_state
      ->3 对不起，请[<tell>]您需要[<refer>]{<where>}的[<what>]。
      ->2 请[<tell>]您需要[<refer>]的[<what>|具体内容]。
      ->1 抱歉，请[<tell>]您需要{<refer>}{(day)|今天|[when]}{(location)|当前城市|<where>}
      的[<what>]。

符号说明：
  |：或者
  []：内部元素出现次数≥1
  {}：内部元素出现次数≤1
  ()：对话管理模块的模板中的变量
  < >：自定义语料中的变量
  句子前的数字：该句子的权重，权重越大句子出现的可能性越大
```

图 2-16　询问天气场景中的句子模板

```
<tell> -> [告诉我|补充|说明|输入]
<refer> -> [查询|知道|获取|收到|了解|咨询]
<where> -> [哪里|何处|什么位置|什么地方|什么城市|哪个位置|哪个区域]
<what> -> [天气|哪方面信息|什么信息|哪方面情况|哪方面内容|何种内容]
<when> -> [哪天|什么时间|哪个时辰|什么时候]
```

图 2-17　询问天气场景中的词汇模板

在实际工程中，基于模板的自然语言生成技术更适用于任务驱动的对话系统，这是由于在任务驱动的对话系统中：

（1）对话管理模块会根据当前的对话状态、用户输入等信息，产生下一步动作相关的信息，也就是会确定自然语言产生模块应该选择的句子模板和可选的词汇模板。

（2）任务驱动的对话系统中的自然语言理解模块需要利用词汇模板、句子模板、有限状态自动机等进行槽位填充（slot filling）的相关工作。

读者学习了第 4 章后，会对上述两点有更深刻的认知。

2.2.4　基于深度学习的自然语言生成

2016 年，微软亚洲研究院刘铁岩团队[12]发表了对偶学习相关的研究，将对偶学习应用于机器翻译领域，其成果可延伸到自然语言处理领域（由于自然语言理解和自然语言生成两项任务在本质上是对偶的，可以考虑使用对偶学习提升自然语言理解和自然语言生成的效果）。

另外，根据前文对基于深度学习的自然语言理解的介绍，端到端框架中的 Decoder 部分可以被理解为自然语言生成的技术环节。针对事先没有完备问答库和对话语料库的情况，使用端到端的生成技术生成对话是这几年的一个发展方向。我们已经知道，端到端算法的思路是使用 Encoder 把离散的数字变成向量化的低维空间的语义表示，根据当前输出决定当前回复的第一个词，然后输出第二个词。

除了上述介绍的利用对偶学习和端到端方法进行自然语言生成，还有另外一种基于深度学习的自然语言生成技术是研究热点。

GAN 在计算机视觉，尤其是图像生成方面取得了令人印象深刻的结果。值得注意的是，从噪声中对抗生成自然语言的研究进展与在图像生成方面的研究进展并不相称，自然语言领域的相关研究仍远远落后于基于似然估计的方法（likelihood based method）。2017 年，包括 *Deep Learning* 一书作者、CIFAR Fellow Aaron Courville（亚伦·库维尔）在内的加拿大研究人员在 arXiv 公布了一项研究[13]，为训练 GAN 得到可以生成自然语言的模型提供了一种直接而有效的方法。

该方法的简单之处在于，通过向判别器提供来自生成器的概率分布序列和对应于真实数据分布的矢量序列（a sequence of 1-hot vectors），强制判别器对连续值进行运算。该方法通过引入简单的不依赖于梯度估计函数（Gradient Estimator）的基准解决离散输出空间问题。实验表明，这种处理方法在一个中国诗词数据集上取得了当前已知的最好效果。作者还在该论文中披露了从无上下文和概率上下文无关文法生成句子的定量结果，以及语言建模的定性结果。该研究的创新之处在于，通过测量模型样本在真实数据分布下的似然对结果进行评估，有别于语言模型一般是通过测量模型下样本与真实数据分布的似然进行评估的既有思路（由于使用 GAN 测量模型本身的似然是不可能的，因此无法使用似然估计的方法）。

2.3　对话管理

在理解对话管理模块的作用时，我们可以将对话管理模块比作聊天机器人的大脑，这一模块的主要任务包括维护更新对话状态和动作选择。**对话状态**是一种机器能够处理的对聊天数据的表征。对话状态中包含所有可能会影响机器

下一步决策的信息，如自然语言理解模块的输出、用户的特征等；**动作选择**是指基于当前的对话状态，选择接下来合适的动作，例如向用户追问需补充的信息、执行用户要求的动作等。举一个具体的例子，用户说“帮我给妈妈预订一束花”，此时对话状态包括自然语言理解模块的输出、用户的位置、历史行为等特征。在这个状态下，系统接下来的动作可能是：

（1）向用户询问可接受的价格，如“请问预期价位是多少”。

（2）向用户确认可接受的价格，如“像上次一样买价值两百元的花可以吗”。

（3）直接为用户预订，“好的，为您预订了价值两百元的康乃馨和红玫瑰送给您的母亲。”对话系统输出更新后的对话状态及一个或多个经选择的状态。

对话管理模块负责协调聊天机器人的各个模块，起到维护人机对话的结构和状态的作用。对话管理模块涉及的关键技术包括对话行为识别、对话状态识别、对话策略学习及对话奖励等。

1．对话行为识别

对话行为是指预先定义或者动态生成的用户对话意图的抽象表示形式。对话行为分为封闭式和开放式两种，所谓封闭式对话行为是指将对话意图映射到预先定义好的对话行为类别体系，通常应用于特定领域或特定任务的对话系统，如设置闹钟、票务预订、酒店预订等。例如，“帮我给妈妈预订一束花”可以被标记为 Reservation（Flower_Mom）的对话行为。开放式对话行为没有预先定义好的对话行为类别体系，基于对话行为动态生成对话意图，常用于开放域对话系统，如闲聊系统。例如，“今天真开心啊”这句话对应的对话行为可以通过隐式的主题、*N* 元组、相似句子簇、连续向量等形式表达。

2．对话状态识别

对话状态与对话的上下文（对话的时序）及对话行为相关，在某时刻的对

话行为序列即为某时刻对应的对话状态。因此，某一时刻对话状态的转移由其前一时刻的对话状态与该时刻的对话行为（该时刻的用户输入）共同决定。

3．对话策略学习

对话策略学习采取的方法是让机器从“人—人”的真实对话数据中学习对话的行为、状态信息等，进而使用学习的结果指导机器在“人—机”对话过程中进行策略的选择。一般来说，对话策略学习通过离线的方式进行，即预先让机器进行对话策略学习，然后在对话过程中直接使用对话策略学习的学习结果。

4．对话奖励

对话奖励可以被看作一种评价对话系统效果的评价机制，对话奖励通常将槽位填充效率、回复流行度等参数纳入考量，基于强化学习的长期奖励机制[5]在 2016 年被提出并得到重视。

常见的对话管理方法主要有 4 种。

第 1 种是基于有限状态自动机（Finite State Machine，FSM）的对话管理方法。这种方法需要人工显式地定义出对话系统可能出现的所有状态，当对话管理模块接收到新的输入时，对话状态都会根据输入在预定的状态间进行跳转。当对话状态跳转到下个状态后，该状态对应的动作会被对话系统执行。基于有限状态机的对话管理的优点是简单易用，缺点是状态的定义及每个状态下对应的动作都要靠人工设计，因此难以应用于复杂场景。

第 2 种是基于统计的对话管理方法。简单来说，它将对话过程表示成一个部分可见的马尔可夫决策过程（对话管理模块的输入存在不确定性，因此决策过程为部分可见）。例如，自然语言理解模块输出的结果可能出错，因此对话状态不再是特定的马尔可夫链中特定的状态，而是一个针对所有状态的概率分布。我们设定系统在每个特定状态（state）下执行某一特定动作（action）都会

获得对应的回报（reward）。在整个决策过程中，系统在每个对话状态下选择下一步动作的策略，即选择期望回报最大的那个动作。这种方法的优点是只需定义马尔可夫决策过程中的状态和动作，机器可以通过学习得到不同状态间的转移关系，且可以使用强化学习的方法在线学习出最优的动作选择策略；相对地，这种方法仍然需要人工定义对话系统的状态，因此该方法在不同领域中的通用性不强。

第 3 种是基于神经网络的对话管理方法。这种方法直接使用神经网络学习动作选择的策略，即将自然语言理解的输出及一些其他特征都作为神经网络的输入，而将选择的动作作为神经网络的输出。这样一来，对话状态便可以由神经网络的隐向量表征，也就不再需要人工显式地定义对话状态。在实际应用中，基于神经网络的方法需要大量的训练数据，且其实际效果并未获得大规模应用验证。

第 4 种是基于框架的对话管理方法。这里的框架是指“槽—值”对，框架根据用户输入进行槽位填充，且可以通过规则明确规定在特定槽状态下的用户动作对应的系统动作，这种方法难以延拓至其他领域且无法处理不确定的对话状态，因此经常被应用于特定领域的对话系统。

对话管理面临如下 3 个挑战。

（1）手工编写的对话策略难以涵盖所有对话场景。

（2）基于统计的方法和基于神经网络的方法需要大量对话数据。

（3）需要大量的领域知识、对话知识和世界知识（world knowledge）来生成有意义的回复语义表征。

为了解决上述问题，很多新的方法被提出：One-shot Learning 和 Zero-shot Learning 旨在从少量样本中进行训练，或者在无任何样本的情况下进行信息补

全，以解决对话系统的“冷启动”问题。基于深度的强化学习在对话管理领域主要被用于帮助系统在实际交互中通过最大化回报函数（reward function）学习在特定状态下采取哪种回复，从而不断增强对话模型中的优势策略，削弱负面策略的影响。这样一来，用户会觉得系统越来越人性化、个性化。SeqGAN 采用对抗网络实现了离散序列数据的生成模型，解决了 GAN 难应用于自然语言处理领域的问题，并且可以被用来选择最优的奖励函数及其参数。

2.4 参考文献

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018.

[2] Yu Wu, Wei Wu, Chen Xing, et al. Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-based Chatbots. ACL 2017.

[3] I. V. Serban, A. Sordoni, Y. Bengio, et al. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. pp. 3776-3784.

[4] J. Li, W. Monroe, T. Shi, et al. Adversarial Learning for Neural Dialogue Generation, 2017.

[5] J. Li, M. Galley, C. Brockett, et al. A Persona-Based Neural Conversation Model, 2016.

[6] P. Shvaiko, J. Euzenat, Ontology Matching: State of the Art and Future Challenges. IEEE Transactions on Knowledge and Data Engineering 25.1(2013): 158-176.

[7] M. H. Seddiqui, M. Aono, An Efficient and Scalable Algorithm for Segmented

Alignment of Ontologies of Arbitrary Size, Web Semantics: Science, Services and Agents on the World Wide Web, vol. 7, no. 4, pp. 344-356, 2009.

[8] W. Hu, Y. Qu, G. Cheng, Matching Large Ontologies: A Divide-and-Conquer Approach, Data & Knowledge Engineering, vol. 67, no. 1, pp. 140-160, 2008.

[9] S. Duan, A. Fokoue, O. Hassanzadeh, et al. Instance-Based Matching of Large Ontologies Using Locality-sensitive Hashing. pp. 49-64.

[10] J. Li, Z. Wang, X. Zhang, et al. Large Scale Instance Matching via Multiple Indexes and Candidate Selection, Knowledge-Based Systems, vol. 50, pp. 112-120, 2013.

[11] C. Shao, L.-M. Hu, J.-Z. Li, et al. RiMOM-IM: A Novel Iterative Framework for Instance Matching, Journal of Computer Science and Technology, vol. 31, no. 1, pp. 185-197, 2016.

[12] Xia Y, He D, Qin T, et al. Dual Learning for Machine Translation. 2016.

[13] Rajeswar S, Subramanian S, Dutil F, et al. Adversarial Generation of Natural Language. 2017:241-251.

3 问答系统

3.1 问答系统概述

问答系统是信息检索系统的一种高级形式，它通过 Web 搜索或链接知识库等方式，检索到用户问题的答案，并用准确、简洁的自然语言回答用户。本书第 2 章简要阐述了问答系统、对话系统和闲聊系统的区别与联系。问答系统更接近信息检索中的语义搜索，针对用户用自然语言提出的问题，通过一系列的方法生成问题的答案，但与信息检索系统的不同在于，问答系统根据用户的问题直接给出精准的答案，而不是给出一系列包含候选答案的页面。系统生成答案的过程虽然也涉及简单的上下文处理，但通常是通过**指代消解**和**内容补全**完成处理操作的。问答系统主要针对特定领域的知识进行一问一答，侧重于知识结构的构建、知识的融合与知识的推理。

问答系统在任务上与很多相关领域的任务有共同点。例如，问答系统与信息检索均需要根据用户提出的问题在 Web 上进行答案信息的检索，问答系统与

数据库查询（Database Query）均需要在数据库或知识库上进行答案信息的查询。但问答系统与信息检索、数据库查询又有所不同，下面是三者各自的特点及各自适用的场景。

三者特点的对比如下。

1．信息检索

（1）以关键字作为输入，以文档或结构化的数据作为输出。

（2）用户需要让搜索引擎“明白”搜索意图。

（3）想要获得令人满意的信息可能要依赖多种检索操作。

（4）信息检索是一个回复驱动的信息获取过程（Answer-driven Information Access）。

2．数据库查询

（1）以结构化的查询语句为输入，以数据记录（data record）或数据聚合（aggregation）等为输出。

（2）用户需要预先理解数据库的模式和数据库查询语言的语法。

（3）令人满意的查询结果可能依赖于多次查询操作。

3．问答系统

（1）以自然语言问题为输入，以准确的答案为输出。

（2）让机器承担更多数据解释的工作。

（3）问答系统是一个问题驱动的信息获取过程（Query-driven Information Access）。

三者适用场景的对比如下。

1．信息检索

适用于简单信息的获取，问题可以用简单的关键字概括，并且网络上有大量相关的文档可供参考。

2．数据库查询

适用于问题规模小而集中，仅存在少量语义异构信息的场景，这类场景对精确率和召回率的数值要求较高。

3．问答系统

适用于特殊而复杂的信息需求，可以从多样化的、非结构化的信息中获取问题的答案，并且需要对问题进行更多自动化的语义理解。

现有的问答系统根据其问题答案的数据来源和回答的方式的不同，大体上可以分为以下 3 类。

1．基于 Web 信息检索的问答系统（Web Question Answering，WebQA）

WebQA 系统以搜索引擎为支撑，理解分析用户的问题意图后，利用搜索引擎在全网范围内搜索相关答案反馈给用户。典型的系统有早期的 Ask Jeeves 和 AnswerBus 问答系统。

2．基于知识库的问答系统（Knowledge Based Question Answering，KBQA）

KBQA 系统通过结合一些已有的知识库或数据库资源（例如 Freebase、DBpedia、Yago、Zhishi.me 等），以及利用如维基百科、百度百科等非结构化文本的信息，使用信息抽取的方法提取有价值的信息，并构建知识图谱作为问答系统的后台支撑，再结合知识推理等方法为用户提供更深层次语义理解的答案。

3．社区问答系统（Community Question Answering，CQA）

CQA 系统也叫基于社交媒体的问答系统，例如 Yahoo! Answers、百度知道、知乎等问答平台。大多数问题的答案由网友提供，问答系统会检索社交媒体中与用户提问语义相似的问题，并将答案返回给用户。

上述 3 类问答系统中，KBQA 是当下应用最广泛的，该类系统不仅需要实现对复杂问题的语义理解，还要在若干知识库之间进行知识的融合，并针对复杂的问题进行知识推理。3.2 节将详细介绍 KBQA 中用到的相关技术，3.3 节将介绍如何实现一个简单的问答系统。除了这 3 类主流问答系统，还有其他形式的问答系统，例如混合式问答系统（Hybrid QA）、多语言问答系统（Multilingual QA）、基于常见问题库的问答系统（Frequently Asked Question，FAQ）。

目前，国际上已经有了一些得到商业应用的问答系统，如 Facebook 的 GraphSearch 可以根据用户的自然语言需求，找到与问题相符的信息返回给用户；IBM 的 Watson 系统，是一个针对特定领域的专业知识进行问答的系统，基于自然语言处理和机器学习算法，Watson 系统能够模拟人思考和决策问题的过程，进行理解、推理、学习和交互，并被广泛应用于医疗、金融等行业，辅助专家提供更好的解决方案；苹果的 Siri 则是实现个人助理功能的问答系统中最具代表性的产品之一，Siri 可以通过问答的形式为用户提供打电话、订餐、订票、放音乐等诸多服务。近年来，还有很多公司推出了类似个人助理的问答系统产品，例如微软 Cortana、Viv、Google Now、出门问问等。

目前，一些基于搜索引擎的问答系统也结合了知识图谱的知识，使用语义检索的方式从多种来源收集信息，可以根据用户的问题进行一定的推理，并将适合的答案返回给用户以提高搜索质量，例如 Google 知识图谱和百度知识图谱等。

为了评估问答系统的性能，许多评测任务和评测数据集均得到大家广泛的

重视和使用。根据问答系统适用的语言的不同，各国组织了诸多具有影响力的评测会议，例如，针对英文问答的 TREC QA Track[①]、日语问答的 NICIR[②]、多语言问答的 CLFF[③]及汉语问答的 EPCQA。其中，使用较为广泛的评测数据集有 Free917[④]、WebQuestions[④]、QALD、Simple Questions 等。QALD 的全称是 Question Answering over Linked Data，是多语言的链接数据问答（Multilingual Question Answering over Linked Data，MQALD）系统的评测竞赛活动，其数据来源包括 DBpedia、Yago 和 MusicBrainz。QALD 旨在建立一个统一的评测基准，主要任务分为 3 类：基于 DBpedia[⑤]的多语种问答、基于链接数据的问答，以及基于 RDF 的结构化知识和自由文本数据的 Hybrid QA。WebQuestions 数据集使用 Freebase[⑥]，通过 Google Suggest API 爬取数据，得到候选问题，经筛选最终得到 5810 个问题，利用 Amazon Mechanical Turk 众包服务得到答案（一个问题可能存在多个答案），并利用 Average F1 评价。Free917 数据集同样使用 Freebase，共有 917 个问题，包含 641 个训练样例和 276 个测试样例。

3.2 KBQA 系统

3.2.1 KBQA 系统简介

KBQA 系统是目前应用最广泛的问答系统之一，适用于人们生活的方方面面，例如在医疗、银行、保险、零售等行业建立相应专业知识的问答系统（智能客服系统），可以给用户提供更好的服务。

① https://trec.nist.gov
② http://research.nii.ac.jp/ntcir/workshop/index.html
③ http://nlp.uned.es/clef-qa/
④ https://nlp.stanford.edu/software/sempre/
⑤ https://wiki.dbpedia.org
⑥ https://developers.google.com/freebase/

知识库（Knowledge Base，KB）是用于知识管理的一种特殊的数据库，用于相关领域知识的采集、整理及提取。知识库中的知识源于领域专家，是求解问题所需领域知识的集合，包括一些基本事实、规则和其他相关信息。知识库的表示形式是一个对象模型（object model），通常称为本体，包含一些类、子类和实体。不同于传统的数据库，知识库中存放的知识蕴含特殊的知识表示，其结构比数据库更复杂，可以用来存放更多复杂语义表示的数据。知识库最早被应用于专家系统，它是一种基于知识的系统，包含表示客观世界事实的一系列知识及一个推理机（inference engine），并依赖一定的规则和逻辑形式推理出一些新的事实[1]。

KBQA 是基于知识库中的专业知识建立的问答系统，也是目前最主流的问答系统。常见的知识库有 Freebase、DBpedia 等。知识库一般采用 RDF 格式对其中的知识进行表示，知识的查询主要采用 RDF 标准查询语言 SPARQL。除此之外，还有一些（例如维基百科等）无结构化文本知识库。

虽然不同的问答系统会有不同的体系架构，但一般来说，KBQA 系统包含问句理解、答案信息抽取、答案排序和生成等核心模块，其基本架构如图 3-1 所示。

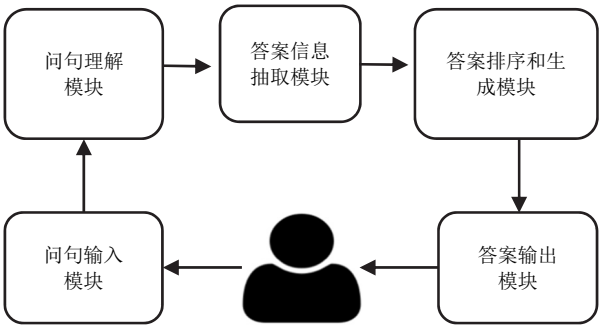

图 3-1　KBQA 系统的基本架构

KBQA 系统中的问句理解模块提取问题中的实体后，答案信息抽取模块通过在知识库中查询该实体得到以该实体节点为中心的知识库子图，并依据某些

规则或模板从提取到的子图中抽取相应的节点或边，得到表征问题和候选答案特征的特征向量，最后将候选答案的特征向量作为分类模型的输入，通过模型输出的分值对候选答案进行筛选，从而得出最终答案。

细化的 KBQA 系统各模块间的关系如图 3-2 所示，其中展示出的主要模块包括问句分析（Question Analysis）、短语映射（Phrase Mapping）、消歧（Disambiguation）和查询构建（Query Construction）。

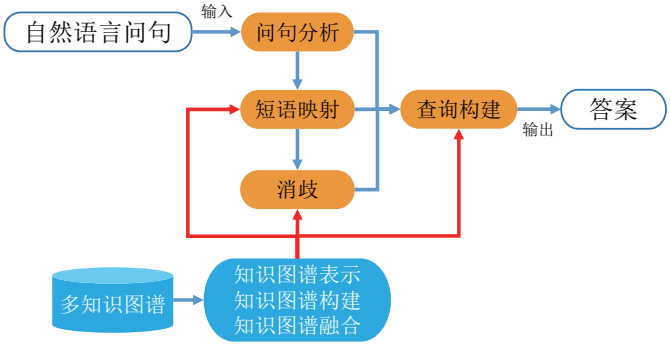

图 3-2 细化的 KBQA 系统各模块间的关系

1. 问句分析模块

提到问句分析，容易联想到自然语言理解，KBQA 系统中用到的问句分析技术属于自然语言处理范畴的任务，但是与自然语言理解技术的侧重点不同，前者更偏重于识别问题中的信息词，例如问题词（谁、什么、何时、事件、为什么、怎么了、如何等）、焦点词（名字、时间、地点）、主题词（可能有多个）、中心动词等词语，也可以理解为前者更集中于实体识别；后者是将自然语言转化成计算机可以理解的形式化语言的过程，包括自动分词（对于中文）、词性标注、命名实体识别、指代消解、句法分析等任务，2.1 节已经介绍了后者相关的内容。这里我们主要结合 KBQA 系统的需要，分析 KBQA 系统中的问句分析。

2. 短语映射模块

短语映射模块主要负责将问题分析模块提取的信息词与知识库或知识图谱中的资源对应的标签映射连接起来。常用的短语映射方法包括本体映射、同义词映射等。在这个过程中，短语映射模块往往通过诸如短语字符串相似度计算、结合外部资源（如 WordNet）进行的词义相似度等语义相似度匹配（Sense-based Similarity Matcher）方法进行相似度计算。

总的来说，相似度计算可以从字符串相似度和语义相似度两个角度进行。字符串相似度计算方法通常使用编辑距离算法、杰卡德距离算法等，也有研究人员使用 Lucene 提供的 FUZZY 模糊查找方法找出与问句信息词最相近的资源标签。关于语义相似度计算的研究很多，也衍生出了许多计算语义相似度的方法，其中较为流行的有如下 3 种。

1）重定向方法

重定向方法基于本体中的 same as 进行映射，或异构本体的锚联结来寻找相同的属性或类，从而扩展问句信息词和标签的映射；利用从语料中抽取的知识找到映射关系，这种知识即语料中自然语言的二元关系，并与本体知识库进行映射。基于重定向方法的语义相似度计算工具有 ReVerb[①]、OLLIE[②]、TextRunner[2]、WOE[3]、PATTY[4]等。

2）使用大型文档找到映射关系

通过大型文档语料库，将本体中的属性（property）进行文字描述的扩展。BOA（Bootstrapping Linked Data）[5]是一个典型的、从大型文档抽取 RDF 模式（pattern）的工具，可以实现属性标签与大规模文档语料中抽取的 RDF 模式建

① http://github.com/knowitall/reverb
② http://github.com/knowitall/ollie

立语义映射关系。

3）基于词向量的方法

根据分布式语义表示的特点，通过词向量计算问句短语词和标签之间的相似度，并对其进行映射，一般采用 Word2vec、GloVe 等工具对自然语言进行向量化。近年来，随着神经网络研究的深入，FastText[①]可以进行快速的词向量训练，ELMo[6]、BERT[7]等模型可以获取高质量的向量化表示，基于获取的词向量可以进行语义相似度计算。

3. 消歧模块

消歧模块又可以理解为候选答案排序（rank）模块。这一模块主要负责解决短语映射模块中出现的歧义问题，以确保问句信息词和知识库实体（资源的标签）的无歧义映射。常用的方法有如下两种：

1）基于字符串相似度的方法

通过计算本体资源的标签和对应的问句信息词之间的相似度进行排序。

2）基于属性和参数的判断方法

通过判断属性和参数（如属性的 domain 和 range）是否一致，去掉不符合一致性的候选答案。具体实现时，可以使用图搜索算法、整数线性规划（Integer Linear Programming，ILP）、马尔可夫逻辑网络（Markov Logic Network，MLN）、结构化感知器（Structure Perceptron，SP）等数学模型，也可以采用人工反馈调整的方法。

① https://github.com/facebookresearch/fastText/

4. 查询构建模块

查询构建模块需要将前面 3 个模块生成的结果进行融合，得到最终的 SPARQL 查询语句，并将查询结果返回给用户。查询模块构建查询 SPARQL 语句的方法可以分为基于模板、基于问题分析、基于机器学习等类型。基于模板构建形式化查询需要预先建立好查询模板，其中包含一些空槽位，将相关信息填入模板槽位后形成一个完整的查询。基于问题分析的方法还可以通过语法树分析、依存树分析或语法槽位等方法，对自然语言进行解析构成查询。同时，还有一些工作是通过机器学习的方法建立问句与查询语句之间映射关系的。

绝大多数的 KBQA 系统都包含以上 4 个核心功能模块，虽然几个模块的顺序可能不尽相同，但不同系统中每个模块完成的功能大体上一致。

典型 KBQA 系统介绍：IBM Watson

目前，工业界已有很多成型的 KBQA 系统，其中最著名的是 IBM 2011 年推出的 Watson 问答系统[①]，它因在美国最受欢迎的智力问答电视节目《危险边缘》中一举打败了人类智力竞赛冠军而名声大噪。本书前述章节已经对 Watson 进行了概要性介绍，本节从技术的角度对 Watson 进行分析。

Watson 采用的知识库是一个广义的知识库，其中不仅包含各种结构化知识，也包含各种非结构化的文本语料和语言学知识。Watson 作为一个集理解、推理、学习、交互功能于一体的强大问答系统，学习处理信息的过程分 4 个阶段，在一定程度上也模拟了人的认知思考过程。

（1）**观察**。观察可见的现象和有形的证据。

（2）**推断**。根据已有知识理解所见之事，然后对其中含义做一些假设。

① https://www.ibm.com/watson/

（3）**评估**。判断某个假设的对错。

（4）**决策**。做出决策，选择最佳选项，并依此采取行动。

整个流程称为 Deep QA，包含问题分解、假设生成、基于证据进行假设评估及排序等关键步骤，这里的 Deep QA 并非指通过深度学习技术提供问答。

图 3-3 所示为 Watson 问答系统的学习过程。首先，通过分析问题的语义，找出查询所需的依赖关系及查询的焦点；然后，根据查询线索生成候选答案，并给出相关性的评分；最后，归并重复的候选答案，由候选答案评估算法做排序选出最终的答案。

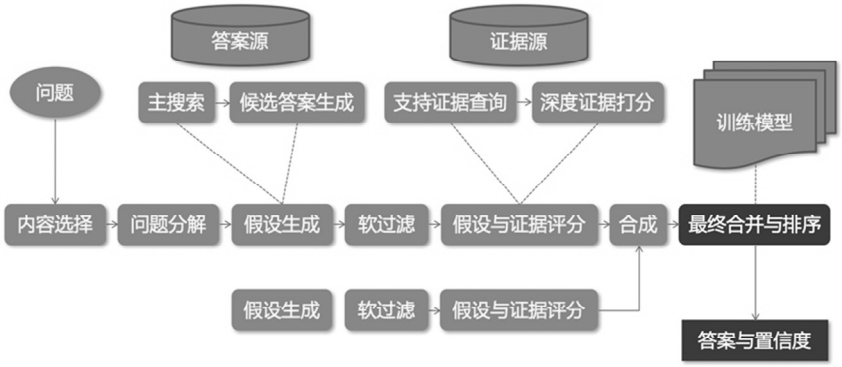

图 3-3　Watson 问答系统的学习过程

当 Watson 在某个特定领域开始工作的时候，它需要学习相应的语言、术语，以及该领域中的思维模式。以癌症为例，癌症有许多类型，每种都有不同的症状和治疗方案，然而除了癌症，其他疾病也可能出现这些症状，因此 Watson 会基于医疗实践和该领域内最优秀的技术文献进行标准评估，从而识别出最佳治疗方案，供医生为患者进行治疗时选择。Watson 的训练需要在“掌握”某个特定领域知识语料库的领域专家的指导下进行。

Watson 的训练过程如图 3-4 所示。首先，进行语料库的“摄取”工作。语

料库包含大量优秀的技术文献，还需加以一定的人工干涉进行降噪，并对数据进行预处理，构建索引和其他元数据，并依此构建一个知识图谱。

然后，对 Watson 进行问答训练。摄取语料库之后，Watson 需要接受人类专家的培训，学习如何理解信息。为了提升 Watson 的学习质量，主要通过机器学习的方法来训练。专家将训练数据以基本问答对的形式输入给 Watson，这里指的并不是问题的明确答案，而是教会它这个领域中专业知识所对应的语言模式。

最后，反馈修正不断学习。接受问答训练以后，Watson 会通过持续交互继续学习，用户和 Watson 之间的交互会定期由专家进行审核，并将反馈输入系统，帮助 Watson 更好地理解信息。新信息发布后，Watson 会根据新信息进行自我更新，以便不断适应特定领域中知识和语言阐释方面的变化。

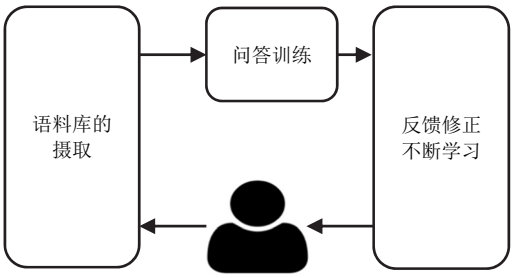

图 3-4　Watson 的训练过程

当用户输入一个问题后，Watson 会经历以下处理过程，才生成最终的答案反馈给用户。

首先，通过自然语言理解处理问句，识别出问题中的一些信息词；接着，Watson 会根据信息词从答案源中生成候选答案，即生成假设；然后，它会寻找支持或推翻每一个假设的论据，并根据每个论据的统计建模结果，对每个论据进行评分，也就是“加权论据得分”；最后，合并每一个假设的所有证据评分并进行综合排序，Watson 会根据答案响应率的高低估计答案的可信度，反馈给

用户。

对 KBQA 的典型产品 Watson 进行技术角度的介绍后，我们将根据问答系统的实现原理，对 KBQA 系统做分类介绍，介绍各类方法的核心思路、代表性系统，以及最新的研究工作。

3.2.2 主流的 KBQA 方法

实现 KBQA 系统的方法（根据其实现原理）可以分为基于模板匹配的方法、基于语义分析的方法、基于图遍历的方法、基于深度学习的方法和其他优化方法。

1．基于模板匹配的方法

1）模板定义

结合知识库的数据结构和问句的句式，对问答系统中的问题进行模板定义。模板定义通常没有统一的标准或格式，需要根据具体的任务需求确定模板的格式。可以参考 Abujabal 等人[8]的定义，将模板格式设定为三元组（U_t, Q_t, M_t）的形式，其中U_t为问题模板，Q_t为查询（query）模板，M_t为问题模板和查询模板之间的映射。也可以参考 Unger 等人[9]的研究，通过定义一个 SPARQL 查询模板将其直接与自然语言映射。

不论是参考定义模板的方式，还是根据具体的需求设计新的模板，模板的定义在基于模板匹配的问答系统实现中起基础作用，对后续根据定义进行模板生成的效果有显著影响。

2）模板生成

根据 Unger 等人[9]研究中对模板的定义，如果要基于该研究进行模板生成，则可以以如下问句为例：

Who produced the most films?

首先，利用词性标注、语法分析、依存分析等方法获得该问句的语义表示，即先将自然语言问句转化为机器可以理解的形式，然后将问句的语义表示转换成相应的 SPARQL 模板：

```
SELECT DISTINCT ?x WHERE {        //要求查询的结果唯一
     ?y rdf:type ?c .             //?y 的类别是电影类
     ?y ?p ?x .                   //?y 是电影，由?x 生产
}
ORDER BY DESC(COUNT(?y))          //对?y 进行计数并降序排序
OFFSET 0 LIMIT 1                  //限制答案数≥0，且≤1，即 0~1。
?c CLASS [films]
?p PROPERTY [produced]
```

有了 SPARQL 模板后，需要将其实例化，也就是将 SPARQL 模板与某一具体的自然语言问句相匹配，填充得到该模板对应的实例，才能查询得到问题的答案。下面就是 SPARQL 模板的一个实例化：

```
?c = <http://dbpedia.org/ontology/Film>
?p = <http://dbpedia.org/ontology/producer>
```

3）模板匹配

模板匹配的过程是将自然语言问句与知识库中的本体概念相映射的过程。在实际操作中，一个问句通常可以匹配到多个模板，同一个模板也可以有多个不同的实例化：

```
SELECT DISTINCT ?x WHERE {
  ?x <http://dbpedia.org/ontology/producer> ?y .
  ?y rdf:type<http://dbpedia.org/ontology/Film> .
}
ORDER BY DESC(COUNT(?y)) LIMIT 1
Score: 0.76

SELECT DISTINCT ?x WHERE {
  ?x <http://dbpedia.org/ontology/producer> ?y .
  ?y rdf:type<http://dbpedia.org/ontology/FilmFestival>.
```

```
}
ORDER BY DESC(COUNT(?y)) LIMIT 1
Score: 0.60
```

针对上述问题的解决方案一般是对每个模板或实例化进行打分，通过排序选择分数最高的答案作为最佳答案，常见的打分排序方法有：

- 对实体等词汇的字符串进行相似度匹配
- 根据模板中槽位填充情况进行打分
- 对实体的属性、类别、领域进行检查

以上便是基于模板方法的问答系统的实现过程。由于自然语言对同一问题的表述千变万化，为了减少人工编写模板的工作量，在实现过程中一般会在基于模板的问答系统中增加模板泛化、自动学习生成新模板等功能。常见的模板泛化方法通常采用同义词替换或基于 WordNet 等外部词典的辅助，使得更多的自然问句可以匹配到系统中已有的模板，也可以对现有的模板进行泛化，自动学习生成新的模板。

综上所述，基于模板的方法的优点在于：

- 模板查询响应速度快
- 准确率较高，可以回答相对复杂的复合问题

其缺点主要集中在以下两方面：

- 人工定义的模板结构经常无法与真实的用户问题进行匹配
- 为了尽可能匹配上一个问题的多种不同表述，需要建立庞大的模板库，耗时费力且查询起来效率较低

2. 基于语义分析的方法

传统的问答系统大多采用基于语义分析的方法进行问句理解，整体思路是

通过对自然语言进行语义上的分析，将其转化成一种知识库真正能“看懂”的语义表示，这种语义表示即逻辑形式（logic form），进而通过逻辑形式访问知识库中的知识，进行推理（inference）和查询，得出最终的答案。

自然语言的逻辑形式表示方法有很多种，这里我们采用λ-DCS[10]（Dependency-Based Compositional Semantics）的表示方法来说明。逻辑形式可以包含知识库中的实体和实体关系（有时也称为谓语或属性），分为一元形式（unary）和二元形式（binary）。对于一个一元实体，我们可以查询出对应知识库中的实体；对于一个二元实体关系，我们可以查到知识库中所有与该实体关系相关的三元组中所包含的实体对。并且，可以像数据库语言一样，对数据进行连接（join）、求交集（intersection）、聚合（如计数、求最大值）等操作。具体来说，可以对自然语言的逻辑形式表示进行以下形式的表示与操作。

一元形式表示：如果实体 $e \in \varepsilon$，那么实体 e 的一元逻辑表示为

$$\|z\|_{\kappa} = \{e\}$$

二元形式表示：如果关系 $p \in \rho$，那么 p 的二元逻辑表示为

$$\|p\|_{\kappa} = \{(e_1, e_2) : (e_1, p, e_2) \in \kappa\}$$

连接操作：如果 b 是二元关系表示，u 是一元关系表示，那么 $b.u$ 表示连接操作

$$\|b.u\|_{\kappa} = \left\{e_1 \in \varepsilon : \exists e_2 .(e_1, e_2) \in \|b\|_{\kappa} \wedge e_2 \in \|u\|_{\kappa}\right\}$$

求交集操作：如果 u_1 和 u_2 都是一元关系，那么 $u_1 \cap u_2$ 表示求交集的操作

$$\|u_1 \cap u_2\|_{\kappa} = \|u_1\|_{\kappa} \cap \|u_2\|_{\kappa}$$

聚合操作：如果 u 是一元关系，那么 $\mathrm{count}(u)$ 表示计数的操作

$$\|\text{count}(u)\|_k = \{|\|u\|_\kappa|\}$$

有了上述定义，我们就可以将自然语言问句表示为可以在知识库中查询的逻辑形式。

最早的基于语义分析的方法使用的是人工编写的逻辑形式规则，人工构建问句的逻辑形式是这一时期的语义分析方法的核心，下面给出的示例是两个句子的逻辑形式规则：

```
What's California's capital?          Capital.California
How long is the Mississippi river?    RiverLength.Mississippi
...                                          ...
```

Berant J 等人[11]研究并公布了将句子建立为语法树的方法，KBQA 系统为问句构建语法树的过程是自底向上构造语法树的过程，这棵语法树的根节点是待分析问句的逻辑形式表达。构造语法树的整个过程可以分为以下步骤。

（1）词汇映射：即构造底层的语法树叶子节点。将单个自然语言短语或单词映射到知识库实体或知识库实体关系所对应的逻辑形式，一般通过构造词汇表（lexicon）来完成词汇映射。

（2）构建语法树：通过自底向上的方式对语法树的节点进行两两合并，最后生成根节点，完成语法树的构建。迄今为止，有很多种构建语法树的方法被提出并得到应用，后续会介绍这些方法中较流行的几种。

图 3-5 所示为句子“What city was Obama born? ”的语法树。

图 3-5 中底层的叶子节点即原始的问句中的词汇，顶层的 Type.City ∩ PeopleBornHere.BarackObama 是构建好的逻辑形式。

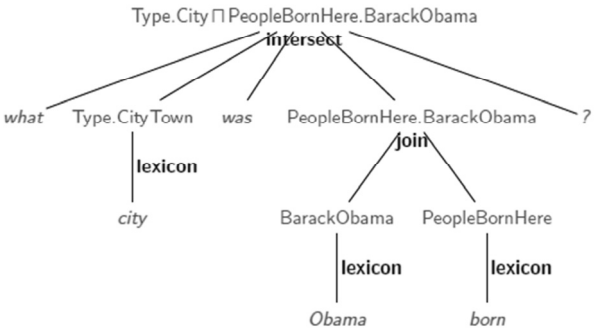

图 3-5　句子的语法树示例

语法树的构建通常包含以下两个步骤。

第 1 步：词汇映射

词汇来源于知识库，想要将自然语言短语或单词节点映射到知识库的实体或实体关系，需要构造词汇表来完成这样的映射。词汇表存放的内容即自然语言与知识库中的实体或实体关系之间的映射，这一操作也被称为对齐（alignment）。一些简单的映射可以采用字符串匹配的方式进行，如图 3-6 所示，将“Obama was also born in Honolulu”中的实体 Obama 映射为知识库中的实体 BarackObama。

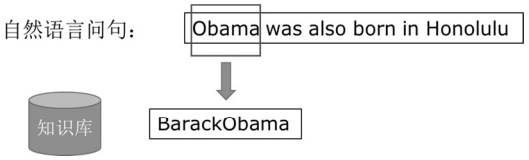

图 3-6　采用字符串匹配的方式进行映射的例子

如图 3-7 所示，要将“was also born in”映射到相应的知识库实体关系，如 PlaceOfBirth，较难通过字符串匹配的方式进行。在这种情况下，可以采用统计的方法，假设文档中有较多的实体对（entity1,entity2）作为主语和宾语出现在“was also born in”的两侧，并且在知识库中这些实体对也同时出现在包含 PlaceOfBirth 的三元组中，那么我们可以认为“was also born in”这个短语可以和 PlaceOfBirth 建立映射。比如(“Barack Obama”, “Honolulu”)、(“MichelleObama”,

“Chicago”)等实体对在文档中经常作为“was also born in”这个短语的主语和宾语，并且它们也都和实体关系 PlaceOfBirth 组成三元组出现在知识库中，因此可以在“was also born in”和 PlaceOfBirth 之间建立映射。将映射方法可视化后如图 3-8 所示。

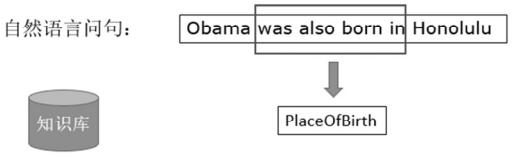

图 3-7 较难通过字符串匹配的方式进行映射的例子

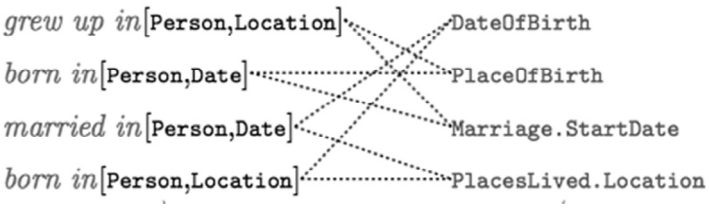

图 3-8 映射方法的可视化示例

在实际工程中，通常通过词性标注、命名实体识别等方式确定哪些短语和单词需要被映射，从而忽略停用词进行词汇映射。另外，还可以建立启发式规则，对问题词（question word）进行逻辑形式的直接映射，如直接将“where”“how many”映射为 Type.Location 和 Count。

第 2 步：构建语法树

自底向上地对语法树的节点进行两两合并，直至生成根节点，完成整个语法树的构建。构建语法树的语法规则有很多，例如组合范畴语法、lambda calculus（λ-calculus）方法[12]、“移位–归约”推导（Shift-reduce Derivation）方法[13]、同步语法（Synchronous Grammar）方法[14]、混合树（Hybrid Tree）方法[15]、类 CFG 语法（CFG-like Grammar）、类 CYK 方法（CYK-like Grammar）、PCFG 语法等。

构造出的语法树即语义分析的结果，逻辑形式就是从构建的语法树中提取出来的。

上述构建语法树的传统方法存在很多局限，例如消耗人才资源多、无法对模板进行快速扩展，且一般需要限定在某一特定领域。为了解决上述问题，研究人员往往采用弱监督学习的方法，根据知识库及问题答案对（question/answers pairs）数据集训练分析器。对于新的问句，通过训练分析器对问句进行语义分析，构建其逻辑形式，进而将问题 x 与答案 y 相映射。问答对的数据集可以从评测比赛中获得，如 QALD、WebQuestions、Free917 等，也可以采用人工的方式从知识库中抽取构建。另外，通常需要对抽取的问答对进行泛化操作，即将原有的一问一答对（q, a）中的 q 进行泛化，衍生出一些表达相同含义的 q_i 的集合 Q，$q_i \in \mathrm{Q}$，即（Q, a）。

通过语义分析构建逻辑形式的具体算法过程如算法 3-1 所示。

算法 3-1　通过语义分析构建逻辑形式的具体算法过程

输入：

Knowledge-base K

问答对训练集合$\{(x_i, y_i)\}_1^n$

问答对示例：

```
What's California's capital?          Sacramento
How long is the Mississippi river?    3,734km
```

输出：

通过语义分析构建逻辑形式，将问题 x 与答案 y 相映射。

```
What's California's capital?  ⇒   Capital.California
                              ⇒   Sacramento
```

知识库中蕴含着丰富的信息及各种关系连接，将其构建成知识图谱，将得

到一个信息量极大的知识网络。知识图谱的可视化表示如图 3-9 所示。

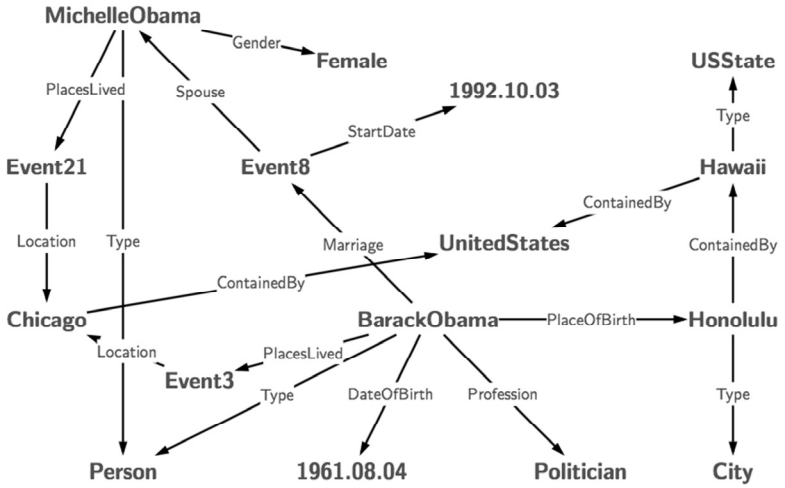

图 3-9 知识图谱的可视化表示

有了图 3-9 所示的知识图谱后，问句的逻辑形式便可以表示成图谱中的一条路径，以 Type.Person ∩ PlacesLived.Location.Chicago 为例，在图 3-10 中进行这一表示的可视化示例。

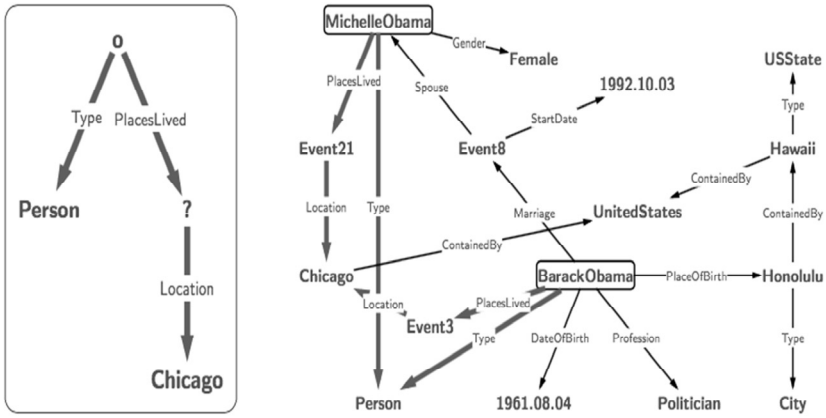

图 3-10 问句的逻辑形式的可视化示例

总的来说，语义分析方法的优点如下：

- 基于人工构建的语法树进行问答，准确率较高
- 对问句的解析深入，因此可以回答相对复杂的复合问题，如时序性的问题

同时，语义分析方法存在以下缺点：

- 需要人工编写大量规则，实现速度慢、人力成本代价高
- 编写的规则模板规模有限，难以跨领域使用

针对以上缺点，下面提供两种优化传统方法的方案。

1）Learning-Based 方法

为了解决问题，Zettlemoyer 和 Collins[16]通过建立统计模型的方式，基于人工预设的学习模板扩展词典；Kwiatkowski [17]提出了一个高阶统一的程序，将大的逻辑形式拆分成小的子部分，他在接下来的工作中[18]还提出了基于因式分解的方法将词典分解成词汇单元和词汇模板；Wong 和 Mooney[19]则假设不同语言的语句逻辑形式具有相同的含义，利用 IBM 翻译模型来学习对应的语句和逻辑形式。

2）神经网络的方法

2015 年，Yih 等人[20]将神经网络的方法加入了语义分析过程，通过在传统的语义分析方法的资源映射过程中融入卷积神经网络，提升单纯语义分析方法的效果。

3. 基于图遍历的方法

最典型的基于图遍历方法的问答系统就是基于图的问答系统，它是基于深度学习方法的前身，这种方法主要可以解决语义词汇映射和歧义这两个问题，将关系抽取转化为图搜索和图遍历过程，显著弱化语义分析方法中关系抽取和

映射的难度。图遍历方法与基于深度学习方法的不同之处在于词汇的映射和候选答案的排序过程。

基于图遍历方法的问答系统的整体框架如图 3-11 所示。

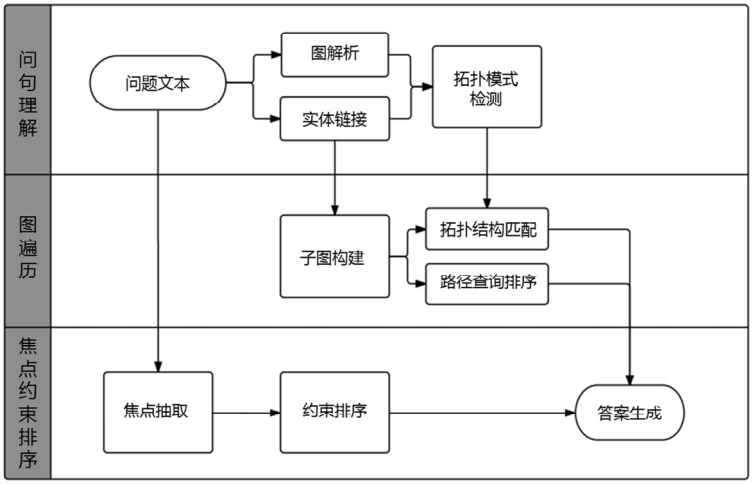

图 3-11　基于图遍历方法的问答系统的整体框架

基于图遍历方法的问答系统在执行的过程中主要有以下 3 个模块。

1）问句理解（Question Understanding）

系统会提取问题中的实体（可以结合规则、模板、依存分析等方法），使用实体链接的方法检测候选实体，并通过建立拓扑模式发现实体的内在关联。

2）图遍历（Graph Traversal）

系统会在知识库或知识图谱中查询该实体，得到以该实体节点为中心的知识库子图（这里的子图中的每一个节点或边都可以作为候选答案），并采用联合排序法（Jointly Ranking Method）遍历图，找到一个最佳路径。

3）焦点约束排序（Focus Constraint Ranking）

系统会抽取可以描述答案的问题核心词，再依据核心词描述生成最终答案。

典型的基于图遍历方法的问答系统产品有 IBM 的 Watson。

4. 基于深度学习的方法

基于深度学习的问答系统采用的是一种基于匹配（Matching-based）的方法。传统方法存在人工编写模板、人工设计语义分析规则、工作量繁重等缺点，随着深度学习方法的流行，自动完成问句理解和知识库映射成为新的研究焦点。

KBQA 与深度学习方法结合主要有两个主流的方法，一种是利用深度学习的方法对传统方法进行改进。例如，利用深度学习的方法进行实体识别、关系识别、实体及实体关系映射（资源映射任务）等。如图 3-12 所示，将图中传统的、采用语义分析处理的部分改用神经网络来操作，可以大大降低人工参与的成本。

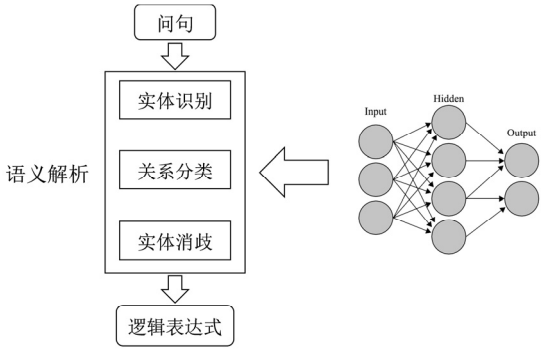

图 3-12　将传统的、采用语义分析处理的部分改用神经网络来操作

另一种方法是采用端到端的策略，在系统中输入问句和知识库，系统直接返回输出答案，中间的操作过程类似于黑盒操作，深度学习被用于候选答案排序的环节。基于深度学习的端到端问答系统的操作过程如图 3-13 所示。

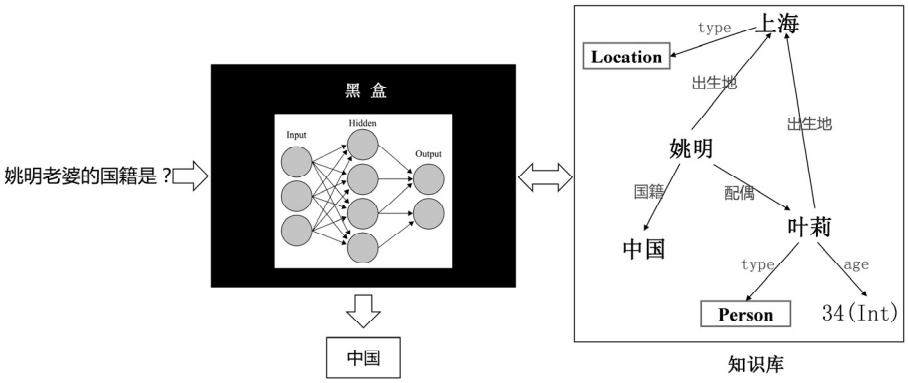

图 3-13 基于深度学习的端到端问答系统的操作过程

下面我们分别介绍这两类方法，以及一些深度学习的高阶方法。

1）利用深度学习改进传统方法

Yih 等人[20]2015 年发布的研究，是一项典型的利用深度学习方法提升传统的单纯语义分析方法效果的工作。该方法将自然语言问题表示成一个查询图的形式，代替了传统的语法解析树的逻辑形式。例如，问句“Who first voiced Meg on Family Guy? ”基于 Freebase 的一个查询图可以表示成图 3-14 所示的形式。

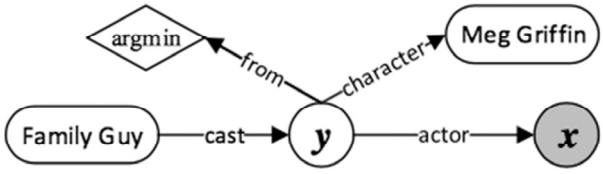

图 3-14 问句的查询图表示示例

其中：

（1）知识库中的实体（Family Guy 和 Meg Griffin），在图中用圆角矩形表示。

（2）存在变量（y），在图中用白底圆圈表示。

（3）聚合函数（argmin），在图中用菱形表示。

（4）λ变量（答案 x），在图中用灰底圆圈表示。

图 3-14 中实体节点到答案变量的路径可以转化为一系列的连接操作，不同路径可以通过交集操作结合到一起。因此，该查询图在不考虑聚合函数最小值的情况下可以转化为一个λ变量的表达式，即

$$\lambda x.\exists y.\text{cast}(\text{FamilyGuy}, y) \wedge \text{actor}(y, x) \wedge \text{character}(y, \text{MegGriffin})$$

上式表示要寻找答案 x，使得在知识库中存在实体 y，满足：

（1）y 和 FamilyGuy 之间存在 cast 关系。

（2）y 和 x 之间存在 actor 关系。

（3）y 和 MegGriffin 之间存在 character 关系。

可以把 y 想象成中间变量，通过对它增加约束来缩小它的范围，并通过它和答案 x 的关系来确定答案 x。有了查询图之后，将其转化为λ表达式，就可以在知识库中查询得到答案了。整个算法的思路归根结底还是语义分析方法的解决思路，但可以利用深度学习方法对构建查询图的过程进行优化。具体的优化方法为，先对问题分析过程中得到的候选主题词进行分析，如图 3-15 所示，在候选主题词和知识库中的实体之间建立映射，并将从被映射实体出发，遍历周围节点长度为 1 的路径（S_5）、长度为 2 且包含 CVT［复合值类型（Compound Value Type），是 Freebase 中可以连接多元实体表示复杂数据关系而引入的概念］节点的路径（如 S_3、S_4）都列为候选路径，成为谓语序列（如 cast-actor 这样的序列）。

采用基于卷积神经网络的方法对候选谓语序列进行打分。将自然语言和谓语序列分别作为输入，分别经过两个不同的卷积神经网络，输出 300 维的分布

式表示，然后可以利用向量间的相似度（如余弦距离）计算自然语言和谓语序列的相似度得分，对候选谓语序列进行打分的过程如图 3-16 所示。

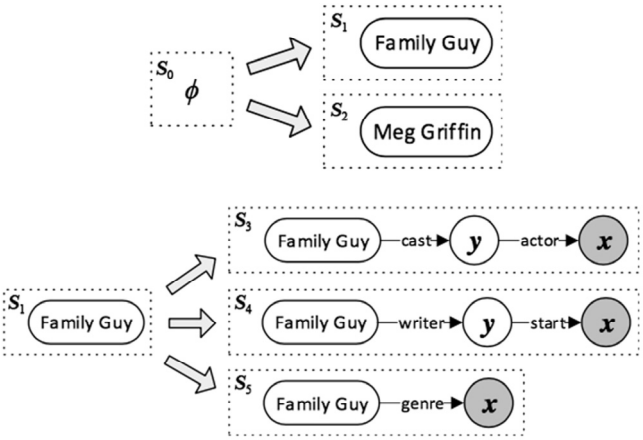

图 3-15　获得谓语序列的过程

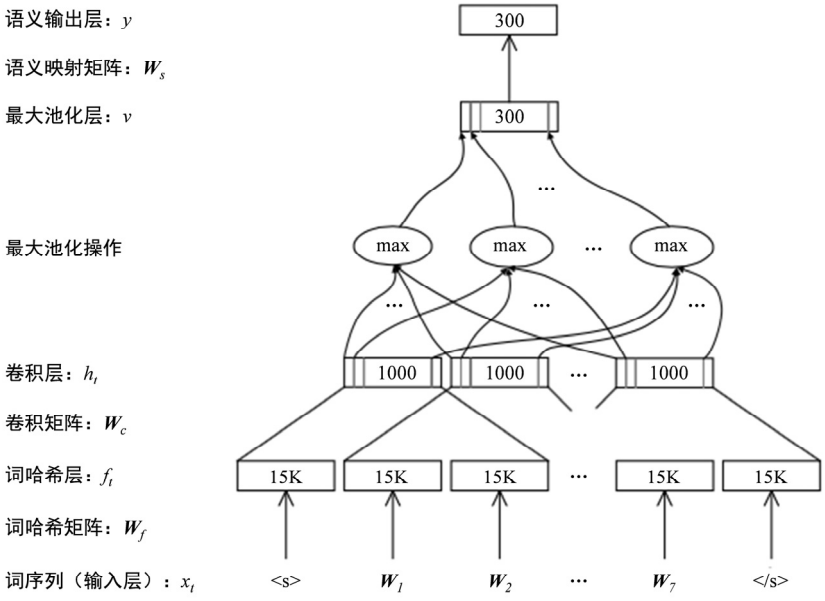

图 3-16　对候选谓语序列进行打分的过程

在实际操作过程中，还可以通过增加约束和聚合函数（如 COUNT、MAX、

MIN 等）提升系统的效果。

这份工作是利用深度学习方法提升传统语义分析方法的一个代表。

2）端到端的深度学习方法

利用端到端的深度学习方法构建 KBQA 系统的典型工作有 2014 年 Bordes 等人发表的研究工作[21]，该工作建立了一个基本的端到端的问答系统，该系统使用已有问答对（主要来自 WebQuestions 和利用 Freebase 提取出的一些新的问答对，包含正负样例等）训练神经网络模型，可以自动地从大规模知识库中学习知识，回答广泛领域主题的问题，同时仅需少量由手工设计的特征。

系统模型的学习过程大致分为以下 5 个步骤，下面以输入问句“Who did Clooney marry in 1987?”为例说明该系统的执行过程。

（1）利用实体链接技术定位问题中的主实体，如“Clooney”。

（2）在知识库中检测出主实体对应的实体表示，在本例中为“G.Clooney”，然后找到从问题实体到答案实体的路径。

（3）将答案实体表示成一个路径，即将知识库中与答案实体有连接的所有实体构成子图，作为候选答案。

（4）将问题和答案子图分别映射成向量，学习出向量表示。

（5）通过点积操作获得问题和候选答案之间的相似度分值。

参考文献[21]中模型的处理过程如图 3-17 所示。

还有其他值得探讨的端到端模型，如针对知识库问答的关系嵌入的相关工作[22]、基于 Freebase 和 CNN 的问答系统[23]，以及基于知识库的端到端问答系统[24]等。

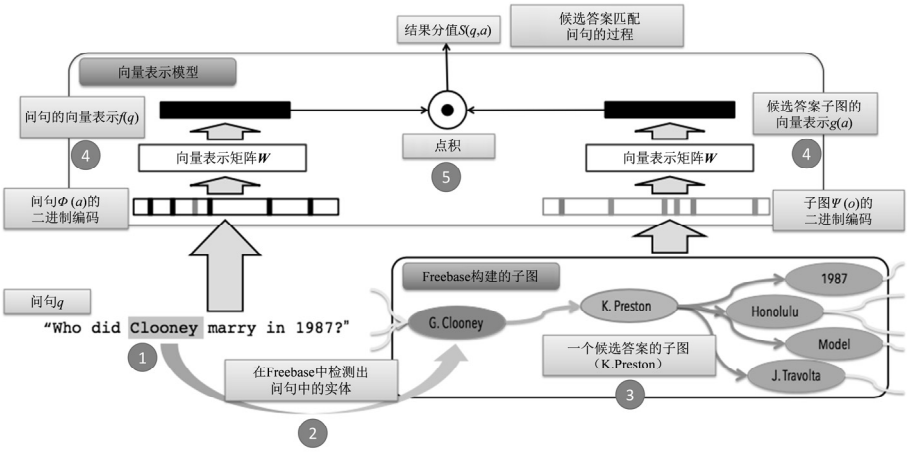

图 3-17　参考文献[21]中模型的处理过程

3）基于深度学习的高阶方法

这里我们引入的记忆网络和注意力机制是问答系统的高阶方法，参考文献[25]为带记忆力机制的深度学习方法，参考文献[24]为基于 LSTM 的带注意力机制的深度学习方法。

总结上述对基于深度学习的方法的讨论，深度学习方法的优点主要集中在无须像模板方法那样人工编写大量模板，也无须像语义分析方法那样人工编写大量规则，整个过程都是自动进行的。其缺点主要有：

- 目前只能处理简单问题和单跳关系问题，处理复杂问题不如两种传统方法效果好。
- 由于深度学习方法通常不包含聚类操作，无法很好地处理时序敏感性问题，例如问句 “who is Johnny Cash’s first wife” 的答案可能是 second wife 的名字。产生错误答案的原因是模型只关注到 wife 而忽略了 first 的含义，并且没有进行额外的推理，而这需要定义专门的操作来优化。

5．其他优化方法

1）多知识库融合

多知识库融合是指为克服单知识库的方法带来的信息不足的缺陷，在实际操作中结合多个不同来源的知识库，进行知识融合的工作。

2）Hybrid QA

Hybrid QA 是指基于知识库和 Web 知识进行的问答，通常是 Web 上半结构化、非结构化的知识，通过 Web 信息检索可以针对知识库信息不全的问题进行补充，这种 Hybrid QA 的方式有其自身的优势，可以覆盖更大的问答范围。

3.3 KBQA 系统实现

本节将以天气领域的 KBQA 系统为例，详细介绍如何设计并实现一个基于知识图谱的问答系统。

3.3.1 系统简介

1．实现目标

系统根据用户输入的与天气相关的问题，理解用户的问题意图，从天气知识图谱数据中检索答案，或加以一定的推理生成候选答案，通过算法进行排序，将最优答案反馈给用户。

2．系统功能

天气问答系统可以回答用户提出的天气相关的一系列问题，其主要功能包括：

（1）回答天气基本信息的问题。例如气温、天气状况、风力风向等。

如：上海今天天气怎么样?

（2）回答天气相关应用场景的问题。例如带伞、洗车、防晒等。

如：今天从上海出门需要带伞吗?

3.3.2　模块设计

问答系统的架构如图 3-18 所示，其有三个核心模块：自然语言理解、查询映射和答案生成。

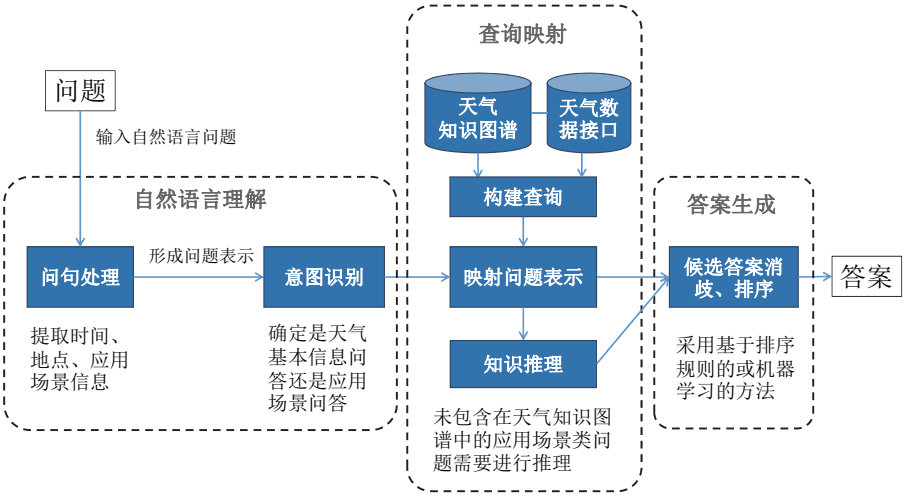

图 3-18　问答系统的架构

自然语言理解模块：也称为问句分析模块，采用模板匹配的方法提取问句中的实体等信息词。该步骤也可以采用自然语言处理领域的技术实现，如中文分词、词性标注、命名实体识别、句法分析等。

查询映射模块：根据自然语言理解模块提取的问句信息和用户意图将自然语言问题转换为相应的查询，进行天气数据接口调用及知识图谱调用。

答案生成模块：候选答案消歧、排序等操作，可以采用基于排序规则的或者机器学习的方法。

部分子模块的功能描述如下。

1. 问句处理模块

该模块的主要任务是识别出问句中的天气信息词，确定问句与天气问答相关，然后提取与天气相关的应用场景词、地域词、时间节点词等。

例 1：今天上海天气怎么样?

time：今天，address：上海，weather_word：天气

例 2：明天从上海出门要带伞吗?

time：明天，address：上海，weather_word：带伞

2. 意图识别模块

确定是问天气基本属性类还是应用场景类问题。根据天气信息词确定咨询的是关于天气的哪一类型的信息，根据是否有场景信息词确定问题属于哪一应用场景。

例 1 中的问句是咨询天气基本信息问题。

例 2 中的问句是咨询天气应用相关的问题：是否带伞。

3. 映射问题表示

（1）用户咨询的问句不一定直接对应知识图谱中的标准表示。例如，知识图谱中存放的是气温字段，而用户咨询的是温度，因此要做词汇映射消歧。

（2）需要映射天气服务接口与知识图谱中的标准表示。

解决映射问题一般采用如下方法：

- 进行字符串相似度匹配（可以采用主流的相似度匹配算法或其他算法）。
- 通过建立同义词表映射解决映射问题。这种方法中同义词表的维护和更新对映射准确度有显著影响。
- 在进行服务接口与知识图谱之间的映射时，可能需要进行必要的拆分和合并操作。

4. 构建查询

该模块通过对输入的问题进行处理，将问题转化为知识图谱查询语言，进而访问知识图谱，通过检索获得答案。这里我们采用 SPARQL 语言访问知识图谱，获得答案信息。

5. 知识推理

如果问题问的是天气基本属性或知识图谱中定义的一些应用场景，则可以从知识图谱中查找，直接返回属性值。

如果询问的是未定义的天气应用场景类问题，则需要通过推理获得答案。

以明天是否需要带伞为例，需要构建的规则样例是：天气状态为下雨则需要带伞，否则无须带伞。

6. 候选答案消歧、排序

从知识图谱中查询到的答案可能不止一个，这种情况下需要对返回的答案进行排序，返回最优的答案。具体模型的设计需要结合具体的领域进行，大致来说，可以选择的方法分为基于规则和基于机器学习两种。

7. 天气知识图谱

将整个天气问答系统看作本体，该本体内部有多个用户查询意图，意图之间也有层次关系，即多级意图。如图 3-19 所示，“天气查询”这个意图为一级意图，而由这个一级意图可以延伸出多个和天气相关的话题，即二级意图，例如洗车咨询、晾晒咨询、防晒咨询、带伞咨询、穿衣咨询等。每个意图又有许多与之密切关联的属性和规则库中对应的规则，例如，在二级意图“防晒咨询”中，一个天气对象名为 weather，与其相关联的属性有：

- 天气对象的温度 weather.temperature
- 天气对象的天气状况 weather.condition
- 查询天气的时间属性 time

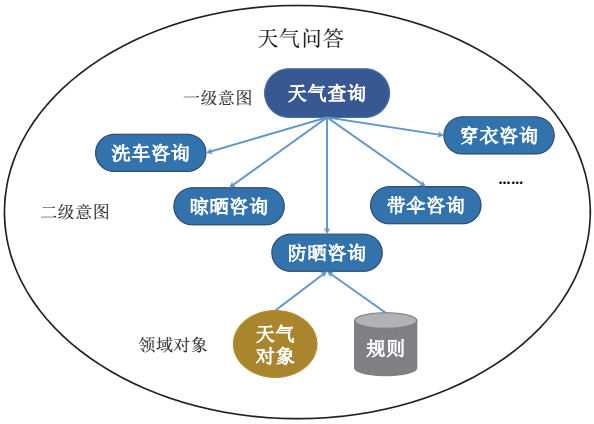

图 3-19　天气知识图谱结构图

上述属性的值可由天气服务接口获取，多个值共同决定防晒咨询意图返回的候选答案。另外，通常还需要执行规则对意图加以约束。例如，当最高气温高于 30 度，天气状况为晴天，且用户咨询天气的时间在 10 点 ~ 16 点的某个时刻时，询问是否需要做好防晒工作，回答应为“是”。

定义好上述天气知识图谱的结构后，天气知识图谱需要与问答系统中的其

他模块产生交互，以便生成最终的答案。天气知识图谱与其他模块交互示意图如图 3-20 所示。

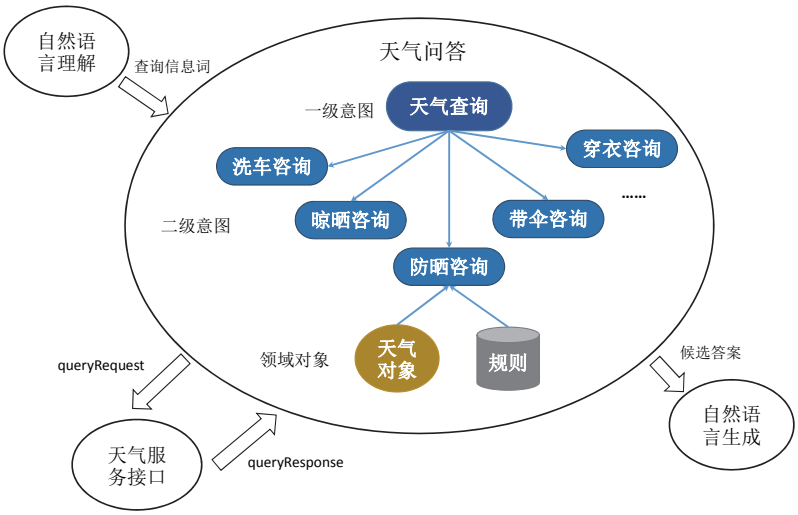

图 3-20　天气知识图谱与其他模块交互示意图

观察图 3-20 可以发现，首先需要由自然语言理解模块进行查询信息词的提取（时间、地点、查询意图词），并将提取到的查询信息词作为输入传递给天气知识图谱。为了将用户的自然语言与天气知识图谱中的标准定义相映射，需要在此时进行本体映射的操作。接着，天气知识图谱会根据用户的意图，向天气服务接口发送一个查询请求（queryRequest），天气服务接口查询到天气知识图谱需要的天气对象信息，将查询响应（queryResponse）返回给天气知识图谱。然后，天气知识图谱将意图所需天气信息及意图对应的规则输出给自然语言生成模块，自然语言生成模块主要做答案生成工作，并对候选答案进行排序，最终将答案反馈给用户。

8. 天气领域问答系统的具体实现

按照上面的模块设计，以结构化的方式构建天气领域问答系统。

系统的核心组成模块共 3 个，即自然语言理解、查询映射和答案生成。下面以伪代码的形式展现系统的实现过程。系统的输入为用户的一次天气问句“sentence”，例如 sentence =“明天上海天气怎么样”，系统的输出为问答系统返回的结果。

天气问答系统的整体算法流程如算法 3-2 所示。

算法 3-2　天气问答系统的整体算法流程

输入：用户输入（sentence）

过程：

```
1.  infoWord ← NLU(sentence);
2.  candidateAnswerList ← MapQuery(infoWord, knowledgeGraph);
3.  reply ← AnswerGeneration(candidateAnswerList);
```

输出：系统回复（reply）

自然语言理解模块由问句处理和意图识别两个子模块构成。这个过程整体采用串行处理方式，先进行问句处理，提取天气问句中的时间（time）、地点（address）、意图词（intent_word）等重要信息词，再根据上述信息词进行意图的分类：根据前述介绍，本例中的意图主要分为两大类，一类是天气基本信息问答，另一类是与天气相关的应用场景问答。特别地，可以通过设计天气信息词字典，以字典数据匹配的方法进行意图识别。天气字典示例如表 3-1 所示。

表 3-1　天气字典示例

意图类别	字典示例
天气基本信息词	天气、天气状况、天气情况、有没有雨、下不下雨 气温、气压、风力、空气质量、污染指数
天气应用场景词	带伞、防晒、紫外线、旅游、钓鱼、运动、晾晒

自然语言理解模块的算法流程如算法 3-3 所示。

算法 3-3 自然语言理解模块的算法流程

输入：用户输入（sentence）
天气字典（weatherDict）

过程：

```
1. [infoWord.time, infoWord.address, infoWord.intent_word] ← questionParsing(sentence, weatherDict);
2. userIntent.type ← intentRecognizer(infoWord.intent_word);
```

输出：问句信息词和用户意图 [infoWord, userIntent]

获取用户问句信息词和用户的意图后，可以根据用户的具体意图和相关约束条件进行数据接口数据查询和天气知识图谱信息查询，生成若干候选答案。这部分模块的输入有 4 个，分别是自然语言理解模块获得的问句信息词（infoWord）和用户意图（userIntent）、外部天气数据或天气数据接口（weatherInterface）、天气知识图谱（weatherKG）。

需要注意的是，如果自然语言理解进行问句处理的结果是用户意图为 NULL，则可能有两种情况：一种是用户输入的不是查询天气的问句，另一种是用户没有给出明确的天气查询意图。在问答系统中，可以直接返回“问句意图不明确，无法查询天气”给用户。当自然语言理解部分返回的用户意图不为空时，才进入查询映射模块，判断时间和地点信息词是否为空；如果为空，直接赋值默认值。例如，可以自行设置 defaultTime，可以是“今天”或者“明天”，defaultAddress 可以是用户所在城市或用户历史查询频率最高的城市。

获得自然语言理解模块的输出后，通过调用 ontologyMappingNL()操作将自然语言理解模块输出的自然语言信息词映射到本体，接着根据用户意图进行问题的分类，根据所需天气信息产生一个 queryRequest 请求，发送给天气服务接口。接着，天气服务接口会将相应的天气信息反馈给系统，系统仍然需要进行一次 ontologyMappingSer()映射操作，以便将天气信息映射到天气知识图谱的标

准表示上，然后系统就可以结合意图对应的规则库里的相关规则，推导出候选答案。查询映射模块的算法流程如算法 3-4 所示。

算法 3-4　查询映射模块的算法流程

输入：问句信息词（infoWord）
用户意图（userIntent）
天气数据接口（weatherInterface）
天气知识图谱（weatherKG）

过程：

```
1.  ifuserIntent !=null then
2.      if infoWord.address ==null then
3.        infoWord.address ← defaultAddress
4.      end if
5.      ifinfoWord.time == null then
6.        infoWord.time← defaultTime
7.      end if
8.      infoWordOntology ← ontologyMappingNL(infoWord, weatherKG.
schema)
9.      weatherInfo ← queryRequest(infoWordOntology, weatherInterface)
10.     weatherInfoOntology ← ontologyMappingSer(weatherInfo,
weatherKG.schema)
11.     candidateAnswer ← generateAnswer(weatherInfoOntology,
weatherKG.rules)
12. else
13.     reply ← “问句意图不明确，无法查询天气”
```

输出：候选答案（candidateAnswer）

查询映射模块获得候选答案后，将候选答案作为答案生成模块的输入。候选答案在该模块中要经过消歧、打分排序等操作，然后系统将获得唯一的答案。最后，系统通过 transformNL()将最终的答案转化为用户可以理解的自然语言表示。

答案生成模块的算法流程如算法 3-5 所示。

算法 3-5 答案生成模块的算法流程

输入：候选答案（candidateAnswer）
问句信息词（infoWord）
答案模板（replyTemplate）

过程：

```
1. candidateAnswerDis ← Disambiguation(candidateAnswer, infoWord);
2. answer ← Ranking(candidateAnswerDis);
3. replyNL ← transformNL(answer, replyTemplate)
```

输出：自然语言回答（replyNL）

问答系统的构建方法相对简单，我们可以在很多具体的方向上进行优化，以提升系统性能。

3.4 参考文献

[1] Hayes-Roth F, Waterman D, Lenat D. Building Expert Systems. 1984.

[2] Banko M, Cafarella M J, Soderland S, et al. Open Information Extraction from the Web. IJCAI. 2007, 7: 2670-2676.

[3] Wu F, Weld D S. Open Information Extraction Using Wikipedia. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2010: 118-127.

[4] Nakashole N, Weikum G, Suchanek F. PATTY: A Taxonomy of Relational Patterns with Semantic Types. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, 2012: 1135-1145.

[5] Gerber D, Ngomo A C N. From RDF to Natural Language and Back. Towards the Multilingual Semantic Web. Springer, Berlin, Heidelberg, 2014: 193-209.

[6] Peters M E, Neumann M, Iyyer M, et al. Deep Contextualized Word Representations. arXiv preprint arXiv:1802.05365, 2018.

[7] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805, 2018.

[8] Abujabal A, Yahya M, Riedewald M, et al. Automated Template Generation for Question Answering over Knowledge Graphs. Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2017: 1191-1200.

[9] Unger C, B ü hmann L, Lehmann J, et al. Template-based Question Answering over RDF Data. Proceedings of the 21st International Conference on World Wide Web. ACM, 2012: 639-648.

[10] Liang P. Lambda Dependency-based Compositional Semantics. arXiv preprint arXiv:1309.4408, 2013.

[11] Berant J, Chou A, Frostig R, et al. Semantic Parsing on Freebase from Question-Answer Pairs. EMNLP. 2013, 2(5): 6.

[12] Yih S W, Chang M W, He X, et al. Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base. 2015.

[13] Zelle J M. Using Inductive Logic Programming to Automate the Construction of Natural Language Parsers. University of Texas at Austin, 1995.

[14] Wong Y W, Mooney R J. Learning Synchronous Grammars for Semantic Parsing with Lambda Calculus. Annual Meeting-Association for computational Linguistics. 2007, 45(1): 960.

[15] Lu W, Ng H T, Lee W S, et al. A Generative Model for Parsing Natural Language to Meaning Representations. Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2008: 783-792.

[16] L.S. Zettlemoyer, M. Collins. Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars, Proc. 21st Conf. Uncertainty in Artificial Intelligence, 2005, pp. 658-666.

[17] Kwiatkowski T, Zettlemoyer L, Goldwater S, et al. Inducing Probabilistic CCG Grammars from Logical Form with Higher-order Unification. Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2010: 1223-1233.

[18] Kwiatkowski T, Zettlemoyer L, Goldwater S, et al. Lexical Generalization in CCG Grammar Induction for Semantic Parsing. Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011: 1512-1523.

[19] Wong Y W, Mooney R J. Learning for Semantic Parsing with Statistical Machine Translation. Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. Association for Computational Linguistics, 2006: 439-446.

[20] Yih W T, Chang M W, He X, et al. Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base. Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing. 2015:1321-1331.

[21] Bordes A, Chopra S, Weston J. Question Answering with Subgraph Embeddings. arXiv preprint arXiv:1406.3676, 2014.

[22] Yang M C, Duan N, Zhou M, et al. Joint Relational Embeddings for Knowledge-based Question Answering. EMNLP. 2014, 14: 645-650.

[23] Dong L, Wei F, Zhou M, et al. Question Answering over Freebase with Multi-Column Convolutional Neural Networks. Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing. 2015:260-269.

[24] Hao Y, Zhang Y, Liu K, et al. An End-to-End Model for Question Answering over Knowledge Base with Cross-Attention Combining Global Knowledge. Meeting of the Association for Computational Linguistics. 2017:221-231.

[25] Bordes A, Usunier N, Chopra S, et al. Large-scale Simple Question Answering with Memory Networks. Computer Science, 2015.

4 对话系统

4.1 对话系统概述

对话系统是一种人机对话交互的典型应用，按照用途可以分为以下两大类。

1. 开放域的对话系统

主要支持闲聊的对话方式，用户通常不具有明确的目的性。在衡量对话的质量上以用户主观体验为主，在实现上主要为基于海量 FAQ 的检索方式，以及端到端的方式。

2. 面向任务的对话系统

指通过对话系统能够指导用户完成一项特定的任务。对话过程通常具有明确的目的性，主要以任务的完成情况来衡量对话的质量，实现上分为基于规则和基于数据两种方式。

这两种类型的对话系统的主要区别在于是否有明确的目的和任务，这一区别使得系统的优化目标不同，相应地，对话质量衡量指标也不同。与较为随意的开放式闲聊对话系统不同，衡量面向任务的对话系统的对话质量时，至少需要知道用户所指定的任务是否被系统正确完成了。例如，购买火车票的需求以购票成功为最终完成指标，再考察需要多少轮对话能完成购票任务，轮数越少越好。本章的后续内容主要围绕面向任务的对话系统展开，闲聊式对话系统的内容将在第 5 章详述。

此外，面向任务的对话系统与第 3 章介绍的问答系统有所不同，后者多为单轮对话，前者多为多轮对话。面向任务的对话系统需要维护一个用户目标状态的表示，并且依赖于一个决策过程来完成指定的任务，因此比问答系统更加关注对话过程，包括目标状态表示和状态迁移，保证目标状态沿着能够完成任务的方向前进。

我们平时所说的 SDS（Spoken Dialogue System），默认指的是面向任务的对话系统。SDS 能够通过语音交互的方式帮助用户完成特定的任务，并且易于嵌入移动设备及终端（如智能手机的语音助手、车载导航系统等）。这种面向任务的对话系统的典型代表有苹果 Siri、微软 Cortana、谷歌助理、百度度秘等。

对话系统的 3 个关键模块为自然语言理解、对话管理和自然语言生成。其中，自然语言理解技术指的是将机器接收到的用户输入的自然语言转换为语义表示，通常包含领域识别、意图识别、槽位填充 3 个子任务。随后，对话管理模块根据语义表示、对话上下文、用户个性化信息（例如年龄）等找到合适的执行动作，再根据具体的动作，使用自然语言生成技术生成一句自然语言，作为对用户输入的回复。

与问答系统不同，面向任务的对话系统的目标是完成用户所指定的一项特定任务，例如查询天气、订餐等。在真实环境中，这些任务往往较复杂，例如

订餐任务需要的必要信息包括用户地址、用户电话、订餐餐厅、订餐菜品等。用户的单轮请求往往无法提供满足任务完成的充足信息，因此多轮对话是必须的，系统通常采用主动询问缺失信息的策略，以填充必要的槽位。如图 4-1 所示，该例中，用户想要查询今天的天气情况，而必要的“地点”槽位不明，系统便采取了主动交互的方式，向用户询问地点信息。完成所有必要槽位的填充后，系统才算成功完成了用户的天气查询任务。

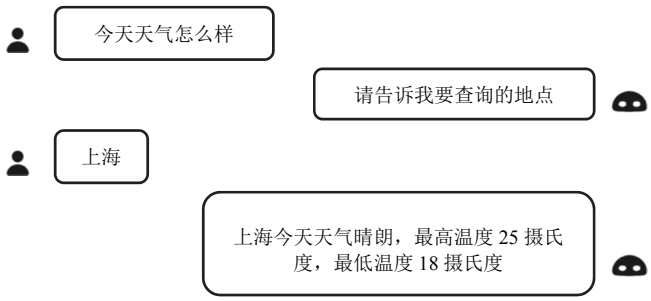

图 4-1　系统采用主动交互的方式，向用户询问地点信息

在训练对话系统时，训练语料是非常关键的，目前可用的语料可参考文献[1]的论述。DSTC（Dialog State Tracking Challenge）是针对任务对话系统中的一部分环节，即“对话状态追踪”模块设立的一个公开评测比赛[2]，其开放的数据评测包含了公交车路线查询、餐厅查询、旅游场景查询几种对话数据，该比赛设立的评测指标很有参考价值（例如，槽位填充的准确度、正确填充所需轮数等）。Ubuntu 对话语料则是 Linux 系统 Ubuntu 论坛社区开发者之间的对话数据[3]，以解决 Ubuntu 系统专业领域问题为主。特别地，在任务型对话语言理解的研究工作中，使用最广泛的是航空旅游信息系统（Airline Travel Information System，ATIS）的数据集[4]，它采集自真实预订飞机票录音。在语言理解任务上使用最广泛的版本是 ATIS-3①，其中包含 4978 句训练数据，893

① https://catalog.ldc.upenn.edu/LDC94S19

句测试数据，127 种语义槽标签和 18 种意图。

因为训练语料可能无法覆盖所有的对话场景，所以系统需要通过反问等方式引导用户补充信息。以图 4-2 为例，除了“锦鲤”这个词，其他内容系统均能够正确地理解。实体“锦鲤”可能由于未被系统的知识库收录或者含有歧义，导致系统无法理解。此时，针对问句中的未识别内容，系统可采用信息补充的主动请求方式，要求用户给出提示，帮助系统理解未识别的内容。在示例中，经过用户的解释，系统成功理解了“锦鲤”的概念，并给出了正确答案。

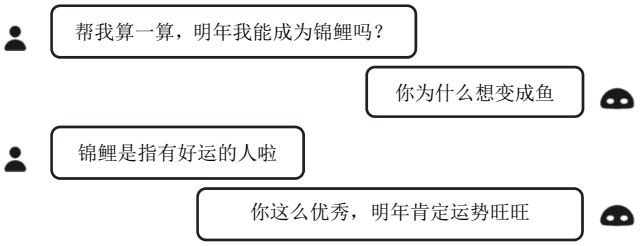

图 4-2　系统采用信息补充的主动请求方式交互

更多关于面向任务的对话系统的综述性介绍可以参考文献[5]和[6]中的内容。

为了方便进一步讲述，接下来对面向任务的对话系统的任务进行形式化定义，如表 4-1 所示。

表 4-1　对话系统符号表

符　　号	解　　释
H_x	用户的对话历史语句
H_y	系统的对话历史语句
X_n	第n轮的用户对话语句
Y_n	第n轮的系统对话语句
u_n	第n轮的用户动作
s_n	第n轮的对话状态
a_n	第n轮的系统动作

给定前$n-1$轮的对话历史信息，包括用户的对话历史语句$H_x = \{X_1, X_1, \ldots, X_{n-1}\}$、系统的对话历史语句$H_y = \{Y_1, Y_1, \ldots, Y_{n-1}\}$及第$n$轮的用户对话语句$X_n$，求$Y_n$。

4.2 对话系统技术原理

按照技术实现，可将任务驱动的对话系统划分为如下两类。

1．模块化的对话系统

分模块串行处理对话任务，每一个模块负责特定的任务，并将结果传递给下一个模块，通常由 NLU、DST（Dialogue State Tracking，对话状态追踪）、DPL（Dialogue Policy Learning，对话策略学习）、NLG 4 个部分构成。在具体的实现上，可以针对任一模块采用基于规则的人工设计方式，或者基于数据驱动的模型方式。

2．端到端的对话系统

考虑采用由输入直接到输出的端到端对话系统，忽略中间过程，采用数据驱动的模型实现。

目前，主流的任务对话系统实现为模块化方式，由于现有训练数据规模的限制，端到端的方式仍处于探索阶段。本章主要介绍模块化的面向任务的对话系统，图 4-3 介绍了其主要模块。

（1）NLU：将用户输入的自然语言语句映射为机器可读的结构化语义表述，这种结构化语义一般由两部分构成，分别是用户意图（user intention）和槽值（slot-value）。

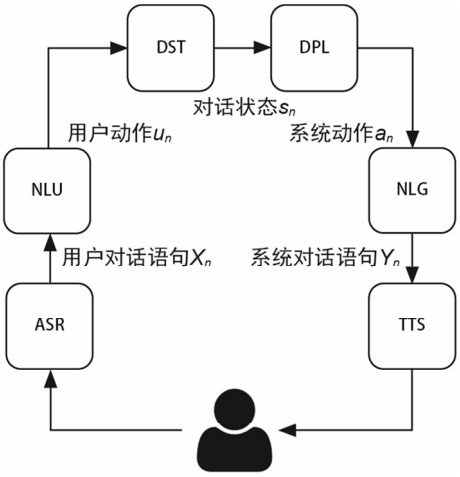

图 4-3　面向任务的对话系统的主要模块

（2）DST：这一模块的目标是追踪用户需求并判断当前的对话状态。该模块以多轮对话历史、当前的用户动作为输入，通过总结和推理理解在上下文的环境下用户当前输入自然语言的具体含义。对于对话系统来说，这一模块有着重大意义，很多时候需要综合考虑用户的多轮输入才能让对话系统理解用户的真正需求。

（3）DPL：也被称为对话策略优化（optimization），根据当前的对话状态，对话策略决定下一步执行什么系统动作。系统行动与用户意图类似，也由意图和槽位构成。

（4）NLG：负责把对话策略模块选择的系统动作转化为自然语言，最终反馈给用户。

在与用户直接关联的两个模块中，ASR 指的是自动语音识别，TTS 指的是语音合成。如第 2 章介绍的，ASR 和 TTS 并不是系统必备的模块，也不是本书介绍的重点，因此在面向任务的对话系统中不对这两部分技术做详细介绍。

4.2.1 NLU 模块

在第 2 章中，已经对 NLU 的功能及研究进展进行了大致的介绍，本节主要结合 NLU 在面向任务的对话系统中的具体应用进行介绍。

对面向任务的对话系统来说，NLU 模块的主要任务是将用户输入的自然语言映射为用户的意图和相应的槽位值。因此，在面向任务的对话系统中，NLU 模块的输入是用户对话语句X_n，输出是解析X_n后得到的用户动作u_n。该模块涉及的主要技术是意图识别和槽位填充，这两种技术分别对应用户动作的两项结构化参数，即意图和槽位。

本节主要讨论如何针对面向任务的对话系统设计 NLU 模块，包括针对特定任务定义意图和相应的槽位，以及后续从用户的输入中获取任务目标的意图识别方法和对应的槽位填充方法。

1. 意图和槽位的定义

意图和槽位共同构成了“用户动作”，机器是无法直接理解自然语言的，因此用户动作的作用便是将自然语言映射为机器能够理解的结构化语义表示。

意图识别，也被称为 SUC（Spoken Utterance Classification），顾名思义，是将用户输入的自然语言会话进行划分，类别（classification）对应的就是用户意图。例如“今天天气如何”，其意图为“询问天气”。自然地，可以将意图识别看作一个典型的分类问题。意图的分类和定义可参考 ISO-24617-2 标准，其中共有 56 种详细的定义[7]。面向任务的对话系统中的意图识别通常可以视为文本分类任务。同时，意图的定义与对话系统自身的定位和所具有的知识库有很大关系，即意图的定义具有非常强的领域相关性。

槽位，即意图所带的参数。一个意图可能对应若干个槽位，例如询问公交车路线时，需要给出出发地、目的地、时间等必要参数。以上参数即“询问公

交车路线”这一意图对应的槽位。语义槽位填充任务的主要目标是在已知特定领域或特定意图的语义框架（semantic frame）的前提下，从输入语句中抽取该语义框架中预先定义好的语义槽的值。语义槽位填充任务可以转化为序列标注任务，即运用经典的 IOB 标记法，标记某一个词是某一语义槽的开始（begin）、延续（inside），或是非语义槽（outside）。

要使一个面向任务的对话系统能正常工作，首先要设计意图和槽位。意图和槽位能够让系统知道该执行哪项特定任务，并且给出执行该任务时需要的参数类型。为了方便与问答系统做异同对比，我们依然以一个具体的“询问天气”的需求为例，介绍面向任务的对话系统中对意图和槽位的设计。

用户输入示例：“今天上海天气怎么样”

用户意图定义：询问天气，Ask_Weather

槽位定义

槽位一：时间，Date

槽位二：地点，Location

“询问天气”的需求对应的意图和槽位如图 4-4 所示。

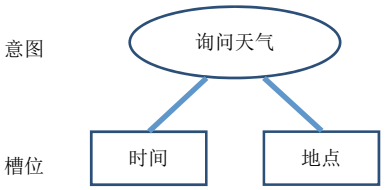

图 4-4　意图与槽位定义（1）

在上述示例中，针对“询问天气”任务定义了两个必要的槽位，它们分别是“时间”和“地点”。

对于一个单一的任务，上述定义便可解决任务需求。但在真实的业务环境下，一个面向任务的对话系统往往需要能够同时处理若干个任务，例如气象台除了能够回答“询问天气”的问题，也应该能够回答“询问温度”的问题。

对于同一系统处理多种任务的复杂情况，一种优化的策略是定义更上层的领域，如将“询问天气”意图和“询问温度”意图均归属于“天气”领域。在这种情况下，可以简单地将领域理解为意图的集合。定义领域并先进行领域识别的优点是可以约束领域知识范围，减少后续意图识别和槽位填充的搜索空间。此外，对于每一个领域进行更深入的理解，利用好任务及领域相关的特定知识和特征，往往能够显著地提升 NLU 模块的效果。据此，对图 4-4 的示例进行改进，加入“天气”领域。

用户输入示例

1.“今天上海天气怎么样”

2.“上海现在气温多少度”

领域定义：天气，Weather

用户意图定义

1. 询问天气，Ask_Weather

2. 询问温度，Ask_Temperature

槽位定义

槽位一：时间，Date

槽位二：地点，Location

改进后的“询问天气”的需求对应的意图和槽位如图 4-5 所示。

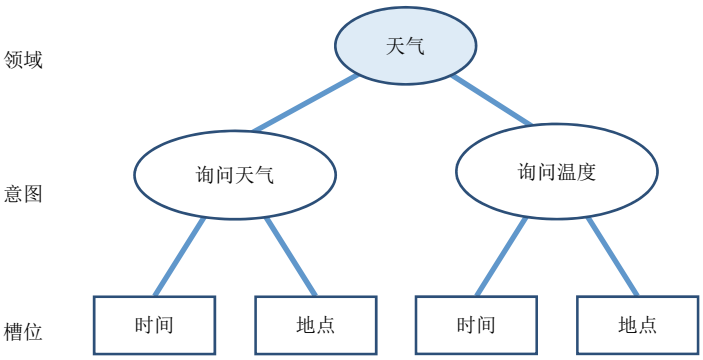

图 4-5　意图与槽位定义（2）

2. 意图识别和槽位填充

做好意图和槽位的定义后，需要从用户输入中提取用户意图和相应槽对应的槽值。意图识别的目标是从用户输入的语句中提取用户意图，单一任务可以简单地建模为一个二分类问题，如“询问天气”意图，在意图识别时可以被建模为“是询问天气”或者“不是询问天气”二分类问题。当涉及需要对话系统处理多种任务时，系统需要能够判别各个意图，在这种情况下，二分类问题就转化成了多分类问题。

槽位填充的任务是从自然语言中提取信息并填充到事先定义好的槽位中，例如在图 4-4 中已经定义好了意图和相应的槽位，对于用户输入“今天上海天气怎么样”系统应当能够提取出“今天”和“上海”并分别将其填充到“时间”和“地点”槽位。基于特征提取的传统机器学习模型已经在槽位填充任务上得到了广泛应用。近年来，随着深度学习技术在自然语言处理领域的发展，基于深度学习的方法也逐渐被应用于槽位填充任务。相比于传统的机器学习方法，深度学习模型能够自动学习输入数据的隐含特征。例如，将可以利用更多上下文特征的最大熵马尔可夫模型引入槽位填充的过程中[8]，类似地，也有研究将条件随机场模型引入槽位填充。基于 RNN 的深度学习模型在意图识别和槽位填

充领域也得到了大量的应用，参考文献[9]介绍了使用 Attention-Based RNN 模型进行意图识别和槽位填充的方法，作者提出将“alignment information”加入 Encoder-Decoder 模型，以及将“alignment information”和“attention”加入 RNN 这两种解决槽位填充和意图识别问题的模型。需要特别介绍的是，与基于 RNN 的意图识别和槽位填充相比，基于 LSTM 模型的槽位填充可以有效解决 RNN 模型中存在的梯度消失问题。

另外，在实际工程中往往需要先对句子中的各个组成部分进行标注，所以通常也会应用到序列标注方法。

进行意图识别和槽位填充的传统方法是使用串行执行的方式，即先进行意图识别，再根据意图识别的结果进行槽位填充任务。这种方式的主要缺陷是：

- 可能产生错误传递，导致错误放大。
- 限定领域也就意味着不同领域需要不同的方法和模型进行处理，各个领域之间的模型没有共享，但在很多情况下，例如订火车票和飞机票时，时间、地点等槽位都是一致的。

因为串行执行的方式存在上述问题，所以研究人员改为使用参考文献[10]设计的联合学习（joint learning）方式进行意图识别和槽位填充。

另外，还有一种情况需要特别注意。在一次天气询问任务完成后，用户又问“那明天呢”时，实际上可以认为第二个问句是开始了另一次“询问天气”任务，只是其中的“时间”槽位是指定的，而“地点”槽位则需要重复利用（继承）上一次任务中的值。

对意图识别模块和槽位填充模块的主要评价指标包括：

- 意图识别的准确率，即分类的准确率。
- 槽位填充的 F1-score。

4.2.2 DST 模块

DST 模块以当前的用户动作u_n、前$n-1$轮的对话状态和相应的系统动作作为输入，输出是 DST 模块判定得到的当前对话状态s_n。

对话状态的表示（DST-State Representation）通常由以下 3 部分构成。

（1）目前为止的槽位填充情况。

（2）本轮对话过程中的用户动作。

（3）对话历史。

其中，槽位的填充情况通常是最重要的状态表示指标。

我们知道，由于语音识别不准确或是自然语言本身存在歧义性等原因，NLU 模块的识别结果往往与真实情况存在一定的误差。所以，NLU 模块的输出往往是带概率的，即每一个可能的结果有一个相应的置信程度。由此，DST 在判断当前的对话状态时就有了两种选择，这两种选择分别对应了两种不同的处理方式，一种是 1-Best 方式，另一种则是 N-Best 方式[11]。

1-Best 方式指 DST 判断当前对话状态时只考虑置信程度最高的情况，因此维护对话状态的表示时，只需要等同于槽位数量的空间，如图 4-6 所示。

图 4-6　1-Best 方式下的对话状态与槽位的对应

N-Best 方式指 DST 判断当前对话状态时会综合考虑所有槽位的所有置信程度，因此每一个槽位的 N-Best 结果都需要考虑和维护，并且最终还需要维护一个槽位组合在一起（overall）的整体置信程度，将其作为最终的对话状态判断

依据，如图 4-7 所示。

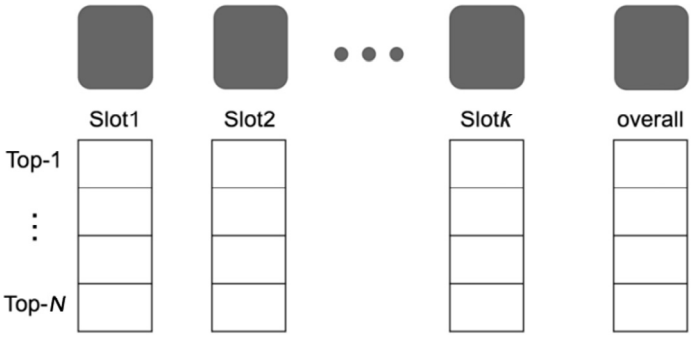

图 4-7　N-Best 方式下的对话状态与槽位的对应

实现 DST 模块的方法主要有：基于条件随机场模型的序列跟踪模型、基于 RNN 和 LSTM 的序列跟踪模型等。

4.2.3　DPL 模块

DPL 模块的输入是 DST 模块输出的当前对话状态s_n，通过预设的对话策略，选择系统动作a_n作为输出。下面结合具体案例介绍基于规则的 DPL 方法，也就是通过人工设计有限状态自动机的方法实现 DPL。

案例一：询问天气

以有限状态自动机的方法进行规则的设计，有两种不同的方案：一种以点表示数据，以边表示操作；另一种以点表示操作，以边表示数据，这两种方案各有优点，在具体实现时可以根据实际情况进行选择。

方案一：以点表示数据（槽位状态），以边表示操作（系统动作）（如图 4-8 所示）

在这种情况下，有限状态自动机中每一个对话状态 *S* 表示槽位的填充情况，例如槽位均为空时，状态为 NULL，表示为(0,0)；仅时间（Time）槽位被填充

时，状态表示为(0,1)。本示例中槽位共有两个，分别为时间和地点（Location），因此共有 4 种不同的状态。

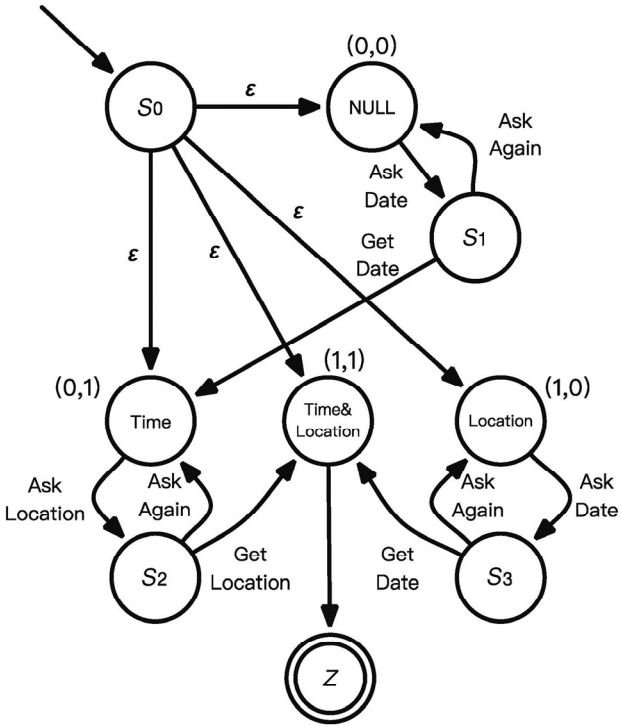

图 4-8 “询问天气”有限状态自动机设计（1）

状态迁移是由系统动作引起的，例如仅时间槽位被填充时，下一步的系统动作为“询问地点”（Ask Location），以获取完整的槽位填充。*S*0 为起始状态，*Z* 为终结状态，*S*1、*S*2、*S*3 三个状态的作用是对槽位填充进行确认。如果成功填充，则跳转到下一个状态继续；如果没有成功，则再一次询问进行槽位填充（Ask Again）。

这种方式的弊端非常明显：随着槽位数量的增加，对话状态的数量也会急剧增加。具体来说，在上述方案中，对话状态的总数由槽位的个数决定，如果槽位有 k 个，那么对话状态的数量为 2^k 个。尝试改进这一弊端的研究有很多，

如 Young S 等人[12]提出的隐藏信息状态模型（Hidden Information State，HIS）和 Thomson B 等人[13]提出的基于贝叶斯更新的对话状态管理模型（Bayesian Update of Dialogue State，BUDS）等。

方案二：以点表示操作（系统动作），以边表示数据（槽位状态）（如图 4-9 所示）

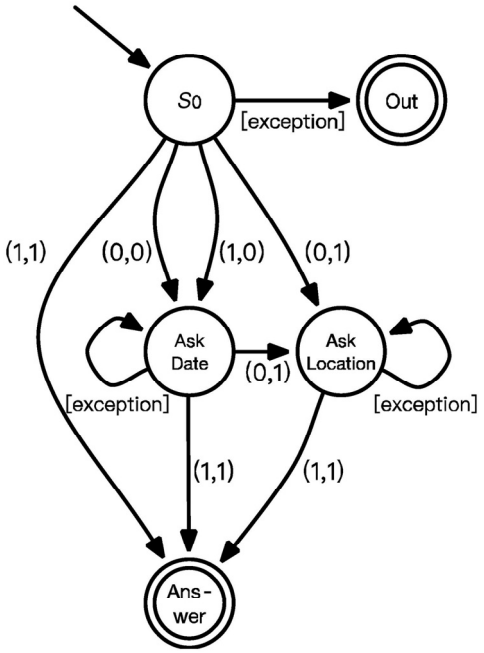

图 4-9 “询问天气”有限状态自动机设计（2）

在这种情况下，有限状态自动机中每一个对话状态 *S* 表示一种系统动作，本例中系统动作共有 3 种，分别是两种问询动作：“询问时间”（Ask Date）和“询问地点”（Ask Location），以及最后的系统回复“回答天气”（Answer）动作。有限状态自动机中状态的迁移则是由槽位的状态变化，即“用户动作”引起的。

对比上述两种方案可以发现，第二种有限状态自动机以系统动作为核心，

设计方式更简洁，并且易于工程实现，更适合人工设计的方式。第一种有限状态自动机以槽位状态为核心，枚举所有槽位情况的做法过于复杂，更适合数据驱动的机器学习方式。

系统动作的定义通常有问询、确认和回答 3 种。问询的目的是了解必要槽位缺失的信息；确认是为了解决容错性问题，填槽之前向用户再次确认；回答则是最终回复，意味着任务和有限状态自动机工作的结束。

细心的读者可能已经发现，采取问询的方式获得缺失的槽位信息，在一些情况下是不合适的，以“询问天气”任务为例，向用户问询槽位缺失的信息会大幅降低用户对系统的满意度。在真实的业务环境下，系统往往会直接采取默认值填充槽位的方式，或者结合以往的对话历史数据，自动填补个性化的结果。例如，用户以往问的都是上海的天气，那么“地点”槽位就会被个性化地填充为“上海”。

这就引出了对面向任务的对话系统的质量评估方法：对面向任务的对话系统而言，完成用户指定任务所需的对话轮数越少越好。在实际应用中，诸如“询问天气”这样的任务，通常都尽可能地在一次对话中完成，而有些任务则必须要进行多轮对话，例如订餐、购票等任务。

接下来，我们以“订餐需求”为例，说明多轮对话的必要性，以及对话轮数的取舍问题。

案例二：订餐

在典型的订餐领域的对话系统中，根据生活经验，我们知道需要为系统定义以下几个槽位。

（1）slot1：用户住址（Address）。

（2）slot2：用户手机号码（Phone）。

（3）slot3：订餐餐厅名称（Res_name）。

（4）slot4：食物名称（Food_item）。

（5）slot5：食物类型（Food_type）。

（6）slot6：价格范围（Price_range）。

其中前 4 项为必要槽位，对订餐任务来说是必须提供的参数，最后两项为非必要槽位，可有可无，有的话可以提高订餐任务的精准度。参考案例一的处理过程，首先对此任务设计相应的有限状态自动机，如图 4-10 所示。

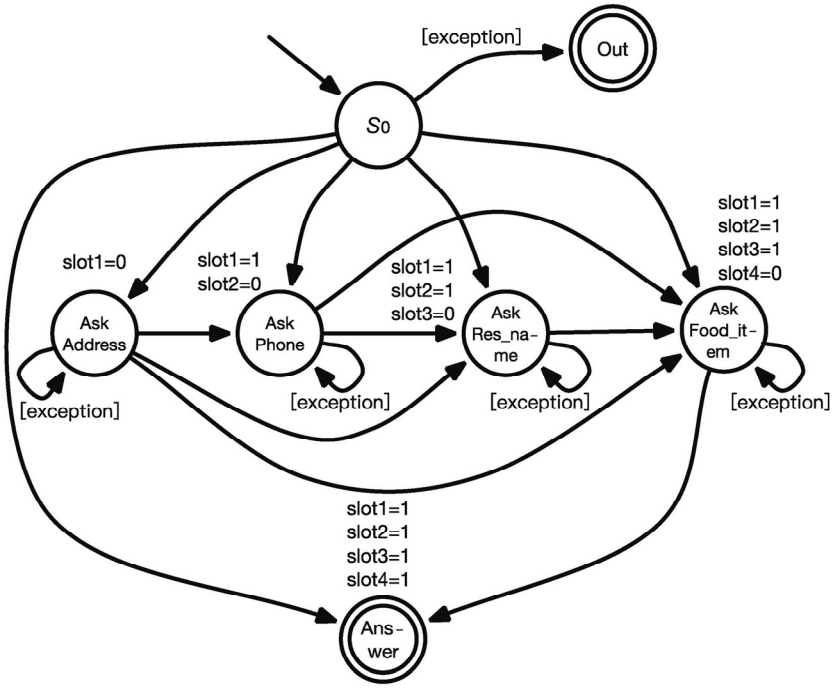

图 4-10　“订餐”系统有限状态自动机设计

可以发现，在该任务下的有限状态自动机设计中，我们只加入了必要槽位

的问询操作，没有对 Food_type 和 Price_range 进行强制的系统问询。两个非必要槽位能够对“Ask Food_item”，即询问具体的食物名称起到辅助作用。当用户没有明确需求时，系统可以给出具体的食物推荐，这样的设定可以有效地减少非必要的对话，减少对话轮数。

4.2.4 NLG 模块

NLG模块的输入是DPL模块输出的系统动作a_n，输出是系统对用户输入X_n的回复Y_n。

目前，NLG 模块仍广泛采用传统的基于规则的方法，表 4-2 给出了 3 个示例规则的定义。根据规则可以将各个系统动作映射成自然语言表达。

表 4-2 NLG 模板规则定义示例

系统动作	系统回复
Ask Date()	“请告诉我查询的时间”
Ask Location()	“请告诉我查询的地点”
Answer(date=$date, location=$location, content=$weather)	“$date，$location 的天气，$weather”

为了实现回复的多样性，各种基于深度神经网络的模型方法被提出并得到发展。

4.3 基于聊天机器人平台搭建对话系统

很多国内外企业和研究单位公开了自己研发的聊天机器人平台，基于这些公开的平台，从业者可以很方便地进行任务驱动的对话系统的搭建，在介绍基于具体开放平台搭建任务驱动的对话系统之前，本节首先对国内外已有的聊天机器人平台进行大致的对比介绍，以方便读者根据具体的项目需求选择适合的平台。

国内已有的开放平台如下。

（1）阿里云提供的智能语音交互[①]平台，主要包括语音识别、语音合成和阿里云人机对话服务，其中阿里云人机对话服务主要提供智能问答和通用领域对话两项服务。通过测试发现，其现阶段主要支持一对一的问答系统，且基于检索的方法实现人机对话。

（2）百度的 AI 开放平台[②]提供的聊天机器人相关的 AI 服务有语言处理基础技术和理解与交互技术（Understanding and Interaction Technology，UNIT），其中语言处理基础技术包括词法分析、依存句法分析、词向量表示、DNN 语言模型、词义相似度、短文本相似度等；UNIT 是百度为第三方开发者提供的对话系统开发平台，适用于智能客服、机器人、智能汽车等应用场景。

（3）ruyi.ai[③]是一个个性化的聊天机器人开放技术平台，支持快速简单定制机器人，协助第三方开发者实现客服、硬件、微信公众号智能化等功能需求，ruyi 提供许多个性化的聊天机器人技能包，并且相对较频繁地推出新的技能包。

（4）IP 梦工厂[④]是一个聊天机器人开放平台，特点是提供个性化 IP 机器人快速定制，预置 20 多种基础性格，例如积极乐观、调皮可爱等。同时，平台原生支持知识图谱，包括 IP 个性图谱、用户画像图谱、百科图谱等，还支持机器人基础问答配置和预置技能包选择。

小 i 机器人[⑤]、图灵机器人[⑥]、竹间智能科技[⑦]等都是国内智能机器人平台和

① 网址：https://data.aliyun.com/product/nls
② 网址：http://ai.baidu.com/
③ 网址：http://ruyi.ai/
④ 网址：https://ipd.gowild.cn
⑤ 网址：http://www.xiaoi.com/index.shtml
⑥ 网址：http://www.tuling123.com/
⑦ 网址：http://www.emotibot.com/

架构的提供者，它们的平台在功能上各有侧重，表 4-3 对国内部分聊天机器人平台的侧重点进行了汇总。

表 4-3　国内部分聊天机器人平台功能汇总

平台名称	问答	对话	槽位提取	技能包	机器人个性设定	记忆	知识图谱
阿里	✓	×	×	×	×	×	×
百度	✓	✓	✓	×	×	×	×
ruyi	✓	✓	✓	✓	✓	✓	×
小 i	✓	×	×	✓	✓	×	×
图灵	✓	×	×	✓	✓	×	×
竹间	✓	✓	×	✓	✓	✓	✓
IP 梦工厂	✓	×	×	✓	✓	✓	✓

国际上也有很多类似的聊天机器人平台，如Amazon的Lex[①]、LUIS[②]、Wit.ai[③]等。LUIS.AI 虽然在系统内置实体数量上不是最多的，但是其支持了自定义特征；而 Wit.ai 不仅在系统内置实体数量上较多，还支持对话管理和对话生成。表 4-4 所示为主流国外聊天机器人平台功能汇总。

表 4-4　主流国外聊天机器人平台功能汇总

平台名称	自然语言理解					对话管理	对话生成
	意图识别	槽位提取	内置实体数量	自定义实体	自定义特征		
LUIS.AI	✓	✓	中	✓	✓	×	×
api.ai	✓	✓	多	✓	×	✓	✓
Wit.ai	✓	✓	多	✓	×	✓	✓
Lex	✓	✓	少	✓	×	✓	✓

聊天机器人平台的评测因素一般需要考虑平台的功能、可用性、效果等。

① 网址：https://amazonaws-china.com/cn/lex/
② 网址：https://www.luis.ai/
③ 网址：https://www.wit.ai/

本节将利用聊天机器人开放平台——百度的 UNIT 平台，以“询问天气”为例，阐述如何从零开始搭建一个解决特定任务的对话系统，并将搭建过程中的每一步与 4.2 节介绍的对话系统框架中的各个模块及方法相对应。

4.3.1　NLU 模块实现

根据 4.2 节的介绍，NLU 模块要实现对自然语言的处理，首先需要针对任务需求进行意图与槽位的定义。由于本例主要以“询问天气”的需求为例进行介绍，可以先将意图定义为“ASK_WEATHER”，即询问天气。参考 4.2 节中对槽位定义的描述与示例，槽位的定义可以同前文示例一致，具体的槽位定义如下。

（1）时间：user_time，自定义时间槽位，可复用平台内置的时间字典。

（2）地点：user_loc，自定义地点槽位，可复用平台内置的地点字典。

图 4-11 给出了 UNIT 平台的槽位定义示例。

图 4-11　UNIT 平台的槽位定义示例

完成意图与槽位的定义后，需要做的是意图识别与槽位填充。意图识别与槽位填充本质上都是通过训练样本进行学习的，所以先导入训练样本数据。如果没有现成的训练数据，则需要开发者手动完成样本数据的建设工作。具体的样本数据准备包括样本的输入与标注两部分，如图 4-12 所示，对于每一句手工输入的样本，开发者都需要进行意图和槽位的标注，分别对应到意图识别任务和槽位填充任务。

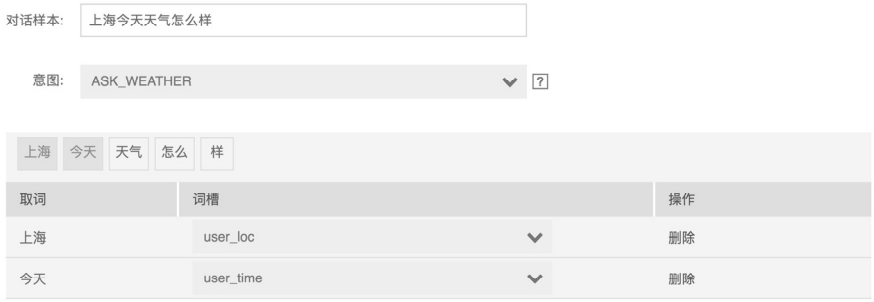

图 4-12　样本标注示例

图 4-13 所示为若干条训练样本最终形成的样本集情况。

对话样本	意图	标注状态
上海今天天气怎么样	ASK_WEATHER	●已标注
上海天气	ASK_WEATHER	●已标注
今天天气	ASK_WEATHER	●已标注
上海今天天气	ASK_WEATHER	●已标注

图 4-13　若干条训练样本最终形成的样本集情况

对于简单的任务对话系统，对话样本数量较少也可以维持基本的运作，如上例中的 4 句对话样本就可以支持“询问天气”任务。但对稍微复杂一点的任务来说，由于复杂的任务往往涉及多个意图，有可能产生冲突导致意图分类出错，这种情况下系统意图识别的准确率就依赖于训练样本的数量和质量。不管是正例样本还是负例样本，均需要足够多的数量，以保证意图识别的准确率。

有了训练数据之后，平台会调用该训练数据进行模型的训练。

4.3.2　DST 与 DPL 模块实现

开放平台的 DST 与 DPL 模块均采用了基于规则的实现方式，因此可以参考 4.2.3 节设计的有限状态自动机来实现，其中有限状态自动机的系统动作可以定义为“Ask Date”和“Ask Location”，即图 4-14 中的“澄清话术”。同时，“澄清顺序”指有限状态自动机中对应槽位均为空时，优先执行“Ask Date”还是“Ask

Location”动作。该例中，与 4.2.3 节有限状态自动机的设计相同，都是优先询问时间，即优先执行“Ask Date”动作。

词槽	描述	澄清话术
user_time	时间	请告诉我查询的时间
user_loc	地点	请告诉我查询的地点

图 4-14 系统动作示例

同时，我们还需要对系统动作“Answer”进行触发条件的设置，如图 4-15 所示。这里将条件设置为槽位填满，即只有当所有槽位均填满时才执行最终的回复命令。

图 4-15 触发条件的设置

可以发现，基于开放平台实现对话系统的核心部分 DST 与 DPL 模块，其设计思路与有限状态自动机的一致，只是用了另外一种形式进行配置，本质上仍然是状态的迁移。

4.3.3 NLG 模块实现

系统回复的实现有两种方式，一种是采用固定文本进行回复，如图 4-16 所示；另一种是通过其他函数或者 API 调用返回动态的结果。以“询问天气”为例，最终需要调用天气查询 API 返回天气数据供对话系统回复给用户。

接下来，以基于 Amazon Lex 的订餐需求定义的任务对话系统的对话效果为例，向读者展示基于开放平台搭建对话任务系统的实现效果，如图 4-17 所示。

图 4-16　采用固定文本进行回复

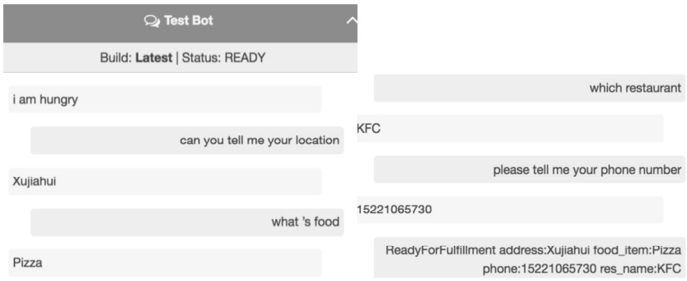

图 4-17　基于 Amazon Lex 的订餐需求定义的任务对话系统的对话效果

4.4　面向任务的对话系统实现

本节主要介绍如何自主实现面向任务的对话系统，如算法 4-1 所示，整体流程由 4 个主要的模块构成，即 NLU、DST、DPL 和 NLG，伪代码的各行分别对各个主要模块的输入和输出进行了定义，例如 NLU 模块的输入为参数 sentence，输出为用户动作 userAct。

算法 4-1　面向任务的对话系统整体流程

输入：用户输入（sentence）

过程：

```
1.  userAct ← NLU(sentence)
2.  dialogState ← DST(userAct, dialogHistory)
3.  systemAct ← DPL(dialogState)
4.  reply ← NLG(systemAct)
```

输出：系统回复（reply）

接下来将对 4 个主要模块分别进行详细描述，需要说明的是，我们仍然以“询问天气”为例演示任务对话系统的实现，并假设用户输入的 sentence 为“今天天气怎么样”。

我们已经知道 NLU 模块由意图识别和槽位填充构成，如算法 4-2 所示，在这里一般采用串行处理的方式，先进行意图识别，再利用意图识别的结果进行槽位填充。意图识别可采用分类算法进行具体的实现，如 SVM。在得到意图之后，槽位填充方法 slotFilling()首先需要获取意图对应的槽位定义，这里的“询问天气”定义了两个槽位，分别是“时间”和“地点”，然后可以采用序列标注法对槽位填充进行具体的实现，如 CRF。意图识别结果和槽位填充结果共同组成了 NLU 模块的输出 userAct。

算法 4-2　NLU 模块算法

输入：用户输入（sentence）

过程：

```
1.  userAct.intent ← intentRecognizer(sentence)
2.  userAct.slotArray ← slotFilling(sentence, userAct.intent)
```

输出：用户动作（userAct）

DST 模块负责接收本轮的用户动作，并判断当前的对话状态。DST 模块在具体实现时认为对话状态同样是由意图和槽位构成的，读者可根据实际情况丰富对话状态的定义。

当意图识别结果不为空时，即正确识别了用户意图时（采用概率的表示时为置信度较高的结果），意味着对话状态进入了一个新的有限状态自动机，即新一轮对话开始，因此对话状态会进行初始化，该示例中我们直接用第一轮对话得到的 userAct 对 dialogState 进行初始化。同时，读者会注意到算法 4-3 中有一个检查默认槽位设定的函数 checkDefaultSlot()，它的作用是对一些槽位进行

默认或者个性化的填充。例如，某一用户的位置在上海，就可以将其询问天气的默认地点槽位个性化地设置为上海，这种设置方式符合人们日常的行为规律，也可有效地减少对话轮数，提高用户对该对话系统的使用体验。

相反，如果意图识别结果为空，则有以下两种情况。

（1）处于多轮对话中，意图和上一轮对话的意图相同。

（2）无未完结的多轮对话，意图识别失败，置为 null。

这种异常情况交由 getIntent()函数进行处理，该意图识别函数需要先考虑历史对话情况，再进行上述两种情况的判断。同时，当处于多轮对话状态时，槽位会被不断填充、更新，槽位更新的工作交由 updateDialogState()函数进行处理，即这个函数负责将本轮获取的槽位更新到整体的历史槽位中。

算法 4-3　DST 模块算法

输入：用户动作（userAct）
　　对话历史（dialogHistory）

过程：

```
1.  if userAct.intent!=null then
2.      dialogState.intent ← userAct.intent;
3.      dialogState.slotArray ← userAct.slotArray;
4.      checkDefaultSlot(dialogState);
5.  else
6.      dialogState.intent ← getIntent(dialogHistory);
7.      dialogState.slotArray ←
updateDialogState(userAct.slotArray , dialogHistory);
```

输出：对话状态（dialogState）

DPL 模块根据当前的对话状态（dialogState）判断下一步的系统动作（systemAct）。如算法 4-4 所示，当对话的意图为“询问天气”时，我们按照算法 4-4 设计的有限状态自动机进行状态判断，分别执行“AskDate”、

“AskLocation”和“AnswerWeather”3 项系统动作。其中，“AnswerWeather”为槽位填满时所执行的系统动作，这部分操作涉及与知识库的连接，并查询指定时间、地点的天气（getWeather()），我们将查询到的天气结果作为槽位填充到系统动作中。当意图为 null 时，系统抛出异常。此外，会话系统往往需要能够处理多项任务，因此可以设计其他意图对应的有限状态自动机并添加到“其他服务”的伪代码位置处。

算法 4-4　DPL 模块算法

输入：对话状态（dialogState）

过程：

```
1.  if dialogState.intent=="询问天气" then
2.      if dialogState.slotArray[0]==null then
3.        systemAct.intent ← "AskDate";
4.      else if dialogState.slotArray[1]==null then
5.        systemAct.intent ← "AskLocation";
6.      else
7.        systemAct.intent ← "AnswerWeather";
7.        systemAct.slotArray[0] ← getWeather(dialogState.slotArray);
8.      end if
9.  else if dialogState.intent==null then
10.     systemAct.intent ← "Exception"; //异常
11. else
12.     其他服务
13. end if
```

输出：系统动作（systemAct）

NLG 模块以套用 NLG 模板的方式实现，如算法 4-5 所示，每一个系统动作（systemAct）对应着一个自然语言表达，作为最后的系统回复输出给用户。这里，我们设计了 4 种系统动作对应的 NLG 模板。

算法 4-5　NLG 模块算法

输入：系统动作（systemAct）

过程：

```
if systemAct.intent=="AskDate" then
    reply ← "请告诉我查询的时间";
else if systemAct.intent=="AskLocation" then
    reply ← "请告诉我查询的地点";
else if systemAct.intent=="AnswerWeather" then
    reply ← systemAct.slotArray[0];
else if systemAct.intent=="Exception" then
    reply ← "抱歉，我刚刚没听清，能再说一次吗";
else
    其他系统动作
```

输出：系统回复（reply）

图 4-18 展示了以上任务对话系统实现后的效果，该示例中，时间、地点槽位均为空，因此系统分别进行了问询，读者可对应上述各模块的伪代码理解。图 4-19 展示了地点槽位为空的情况，系统因此只进行了一次问询。图 4-20 则应用了个性化技术，将地点槽位进行了个性化的默认填充，可以看出，这样的交互非常友善。

图 4-18　系统展示（1）

图 4-19 系统展示（2）

图 4-20 系统展示（3）

4.5 参考文献

[1] Serban I V, Lowe R, Charlin L, et al. A Survey of Available Corpora for Building Data-driven Dialogue Systems. arXiv preprint arXiv:1512.05742, 2015.

[2] Williams J D, Raux A, Ramachandran D, et al. The Dialog State Tracking Challenge, SIGDIAL Conference. 2013: 404-413.

[3] Lowe R, Pow N, Serban I, et al. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-turn Dialogue Systems. arXiv preprint arXiv:1506.08909, 2015.

[4] Raymond C, Riccardi G. Generative and Discriminative Algorithms for

Spoken Language Understanding, Proceedings of the 8th Annual Conference of the International Speech Communication Association. 2007:1605-1608.

[5] Mo K. A Survey of Task-oriented Dialogue Systems. 2017.

[6] Chen Y N, Celikyilmaz A, Redmond W A. Deep Learning for Dialogue Systems. Proceedings of ACL 2017, Tutorial Abstracts(2017), 8-14.

[7] Bunt H, Alexandersson J, Choe J W, et al. ISO 24617-2: A Semantically-based Standard for Dialogue Annotation, LREC. 2012: 430-437.

[8] McCallum A, Freitag D, Pereira F C N. Maximum Entropy Markov Models for Information Extraction and Segmentation, ICML. 2000, 17: 591-598.

[9] Liu B, Lane I. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling. 2016.

[10] Hakkani-Tür D, Tür G, Celikyilmaz A, et al. Multi-Domain Joint Semantic Frame Parsing Using Bi-Directional RNN-LSTM, INTERSPEECH. 2016: 715-719.

[11] Young S, Gašić M, Thomson B, et al. Pomdp-based Statistical Spoken Dialog Systems: A review. Proceedings of the IEEE, 2013, 101(5): 1160-1179.

[12] Young S, Keizer S, Mairesse F, et al. The Hidden Information State Model: A Practical Framework for POMDP-based Spoken Dialogue Management. Computer Speech & Language, 2010, 24(2):150-174.

[13] Thomson B, Schatzmann J, Young S. Bayesian Update of Dialogue State for Robust Dialogue Systems, IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2008:4937-4940.

5 闲聊系统

5.1 闲聊系统概述

闲聊系统与问答系统、面向任务的对话系统三者均为聊天机器人的典型应用，但应用任务目标和实现方式均有较大不同。目前，大量聊天机器人产品定位于闲聊系统，如微软推出的“小冰”。值得一提的是，2018 年 8 月 22 日第 6 代小冰发布，微软宣布小冰逐步进入完成态，实现了从人工智能交互到初级感官再到高级感官的跨越，其核心对话引擎包括情绪识别、兴趣分析、情感策略及主动回应模型，并全面使用了生成模型与用户进行对话。虽然目前对于开放平台的接入还在逐步进行，不过已经可以看到微软小冰在闲聊系统之外的诸多尝试。比较早期的闲聊机器人包括 2013 年的“小黄鸡”，作为一款聊天机器人程序，其上线后在人人网迅速蹿红，三天内累积增长 70 万粉丝，日发送回复量超过 70 万。用户只要在人人网主页@小黄鸡，小黄鸡就会自动回复用户，并与用户聊天。其主要功能是通过将韩国聊天机器人平台 SimSimi 的开放 API 和人人网接口相连实现的。微软小冰也做了网络聊天的尝试，包括其推出的 QQ 版本小冰机器人，以及微博小冰和微信小冰。同时，大量的聊天机器人硬件产品

也基本上都具备闲聊功能，例如小米音箱、天猫精灵、叮咚音箱等。

类似于已经介绍过的问答系统和面向任务的对话系统，根据具体实现方式，闲聊系统也可以分为基于对话库检索的闲聊系统和基于生成的闲聊系统，这两种方法的优缺点在前面章节中均有所涉及。

（1）尽管基于对话库检索的闲聊系统可以有效避免出现语法错误，但很难处理对话库中不存在的或者没有预定义的问题。

（2）尽管基于生成的闲聊系统能比较灵活地整合上下文的信息，但是生成模型的训练需要大量标注数据，且难以避免安全回复的问题和回答中可能出现的不一致问题或语法错误。

无论是基于检索的还是基于生成的方法，都可以在系统中引入深度学习技术。由于端到端的深度学习结构非常适用于文本生成，许多最新的研究工作正试图促进深度学习技术在这个领域取得飞速的进展。但是实际上，由于基于生成的方法还处在发展的早期阶段，其表现并不尽如人意，在实际应用中还是更多地使用基于检索的模型。

5.2 基于对话库检索的闲聊系统

5.2.1 基于对话库检索的闲聊系统介绍

基于对话库检索的闲聊系统指的是事先存在一个对话库，闲聊系统收到用户输入的句子后，在对话库中通过搜索匹配的方式进行应答内容的提取。由于用户在真实场景下对话语料极为丰富，这种方式对对话库中语料的数量和质量要求很高，必须能够尽量多地匹配用户问句。另外，因为对话库中存储的都是真实的问答数据，所以这种方式的回复质量较高，表达比较自然。从本质上讲，

基于检索技术的聊天机器人类似于搜索引擎，其工作流程是事先存储好对话库并建立索引，根据用户输入的内容在对话库中匹配最合适的回复内容。

基于检索的闲聊技术主要使用匹配的方法，而匹配方法的核心是匹配用户问句 ***x*** 和对话库中现有的句子 ***y*** 的相似度并进行排序，选出候选问句。传统的做法是将句子表示成 one-hot 向量，然后对向量求相似度。随着深度学习技术的发展，句子的表示也常采用词嵌入的方式，以便更好地体现句子中的语义信息。

目前主流的匹配方法有两种，一种是弱相关（weak interaction）模型，包括 DSSM[1]、ARC-I[2]等算法，另一种是强相关（strong interaction）模型，包括 ARC-II[2]、MatchPyramid[3]、DeepMatch 等算法。两种方法最重要的区别是对句子<***x***, ***y***>建模的过程不同，前者是单独建模，后者是联合建模。下面将通过几个经典的算法进行阐述。

DSSM 算法采用词袋模型进行句子表示，如图 5-1 所示，Q 表示待匹配的句子，$D_1,\ldots,D_n$ 表示对话库中已有的句子，逐步对句子进行降维，在最后的 128 维向量上做相似度计算，从而选出最相似的句子。这就是很典型的弱相关模型。

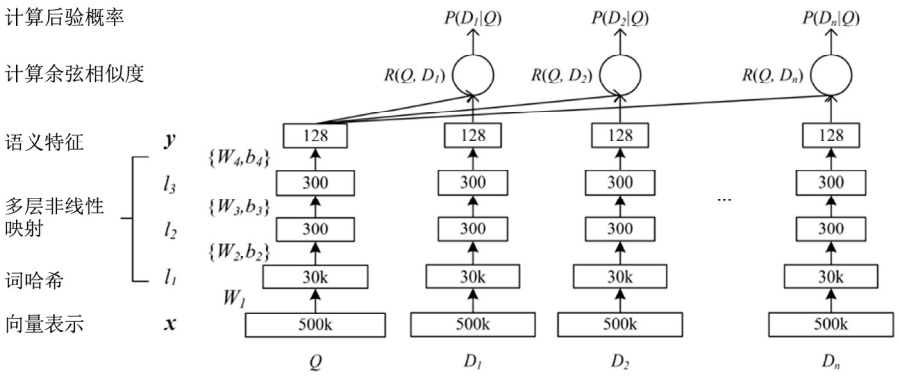

图 5-1 DSSM 算法采用词袋模型进行句子表示

华为诺亚方舟实验室于 2014 年发表了论文[2]，同时给出了两种模型，ARC-I 是弱相关模型，ARC-II 是强相关模型。如图 5-2 所示，ARC-I 算法先对句子单独建模，最后通过一个多层感知机来计算匹配度。

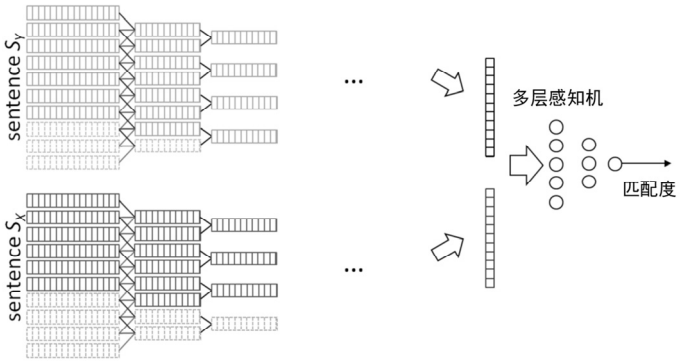

图 5-2　ARC-I 算法

而 ARC-II 算法是将句子的不同词语组合做拼接，再去做更多的卷积和池化，最后得出匹配度，如图 5-3 所示。

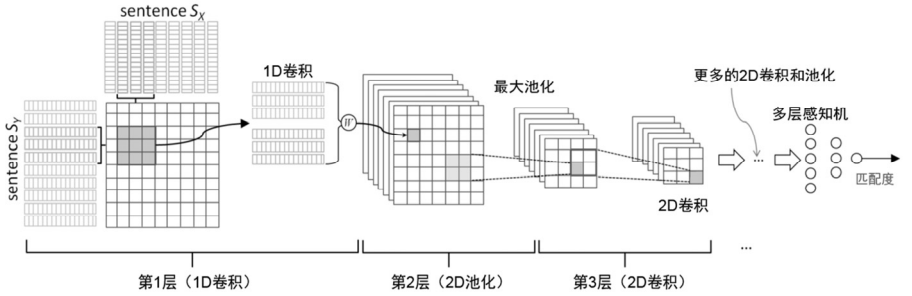

图 5-3　ARC-II 算法

另一个经典的强相关模型是 MatchPyramid，如图 5-4 所示，同样是在一开始就对句子进行联合建模，然后通过多层的卷积得到最终的匹配度。

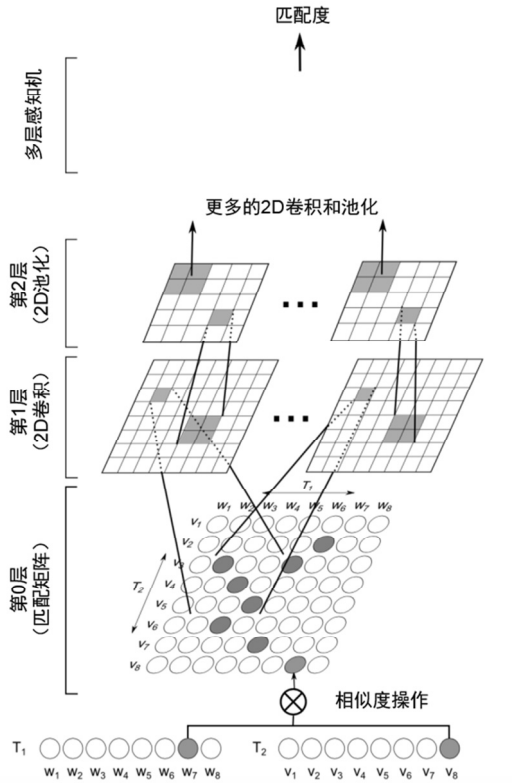

图 5-4 强相关模型 MatchPyramid

5.2.2 对话库的建立

目前，对话库的建立方法有许多种，包括从电影剧本中提取对白、从小说中获取对话内容，以及从网络社区中提取对话库等。举例来说，开放对话库 Ubuntu Dialogue Corpus（UDC）是 Lowe 等人[4]建立的公开数据集，也是目前可用的最大的公共对话数据集之一。该数据集是基于 IRC 网络上 Ubuntu 频道的对话数据和非结构化的社交媒体数据建立的，目前许多聊天系统相关的工作均基于 UDC 数据集进行模型训练和测试。

UDC 1.0 版本包含约 100 万条多轮对话数据，以及超过 700 万条回复和超

过 10 亿个词。UDC 2.0 以时间点为依据将数据划分为训练数据、验证数据和测试数据，让用户使用过去的数据进行模拟训练，以预测未来的数据；且删除了 UDC 1.0 中的分词和指代消解等处理，而用特殊的符号对这些信息进行表示。同时，UDC 2.0 增加了符号表示回复的结束（__eou__）、多轮对话的结束（__eot__）、测试集和训练集的分隔符（__EOS__或</s>）等。

如图 5-5 所示，UDC 中标签为 1 的回答是真正的回答，标签为 0 的回答是从 UDC 中随机挑选出来的语句。

Context	Response	Flag
well, can I move the drives? __EOS__ ah not like that	I guess I could just get an enclosure and copy via USB	1
well, can I move the drives? __EOS__ ah not like that	you can use "ps ax" and "kill (PID #)"	0

图 5-5　UDC 中的数据示例

图 5-6 是 UDC 中句子长度分布的可视化示例，从中可以发现绝大多数的句子是短句，同时大部分对话的长度也较短。

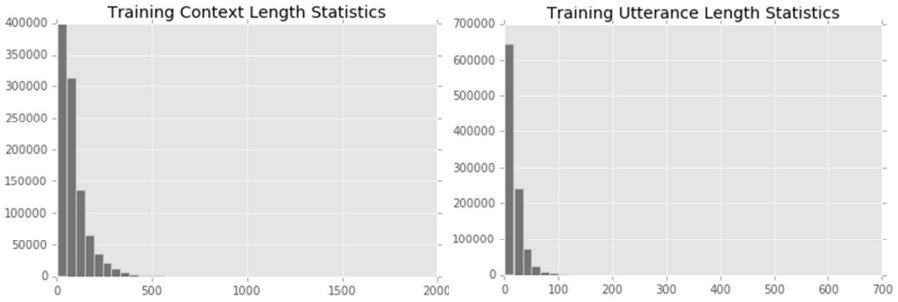

图 5-6　UDC 中句子长度分布的可视化示例

有了对话数据，接下来就可以使用检索的方法，针对用户问句，得到相应的答复。

5.2.3 基于检索的闲聊系统实现

基于检索的闲聊系统的主要设计思想是检索出与当前输入语句最相近的对话库语句,将该语句对应的回复作为系统回复,达到自动生成闲聊回复的目的。因此，检索式闲聊系统的核心为句子的相似度匹配。

为了阐述方便，本节我们将用一个详细的例子介绍基于检索的闲聊系统的实现流程。算法流程大体上包括两个步骤：第 1 步，用一个搜索引擎（如 Elasticsearch[①]）对所有语料进行粗粒度筛选，获得候选答案；第 2 步，使用匹配算法对候选答案进行精排序,获得候选答案中与输入句子语义最接近的问句，返回该问句对应的答句作为最终的回复语句。接下来，我们对这两个步骤进行详细解释。

Elasticsearch 是一个分布式、可扩展、实时的搜索与数据分析引擎，它不仅可以支持全文搜索，还可以支持结构化搜索、数据分析，以及一些更复杂的语言处理、地理位置和对象间关联关系处理等。它可以快速地储存、搜索和分析海量数据，也被维基百科、StackOverFlow、GitHub 采用。Elasticsearch 的底层是开源搜索库 Lucene，其对 Lucene 进行了封装，对外提供 RESTful 风格的 API 接口，使用相当便捷。

Elasticsearch 在计算文本相关度时采用了 Okapi BM25 算法，BM25 算法源自概率相关模型，而非向量空间模型，是对传统的 TF-IDF 算法的改进。介绍 BM25 算法之前，我们首先阐述 TF-IDF 算法的思想和计算公式。

TF-IDF 算法包括两个核心概念，一个核心概念是 TF，它是指一个词在某类文档中出现次数的占比,一个词在文档中出现的次数越多通常说明其越重要。

① https://www.elastic.co/products/elasticsearch

$$\mathrm{TF}(w)=\frac{\text{在某类文档中}w\text{出现的次数}}{\text{该类全部文档中的所有词条数}}$$

另一个核心概念是 IDF，包含某个词条的文档数量越少，说明该词具有区分文档的能力，反之则反。

$$\mathrm{IDF}(w)=\log\left(\frac{\text{语料的文档总数}}{\text{包含词条}w\text{的文档数}+1}\right)$$

其中，分母加 1 是一种平滑方法，避免包含词条 w 的文档数为 0 时，比值无法计算的问题。因此，TF-IDF 的公式为

$$\mathrm{TF}-\mathrm{IDF}=\mathrm{TF}\cdot\mathrm{IDF}$$

BM25 算法同样使用 TF、IDF 及字段长度归一化。与 TF-IDF 不同的是，其增加了可调参数 k_1 和 b。

k_1 代表词频饱和度（Term Frequency Saturation），用来控制饱和度变化的速率和上限。有一些词如“的”“了”等在文档中出现的频次很高，其 TF 值也极高，以致于它们的权重被过分放大。传统的 TF-IDF 在计算时，通常会去掉这些词（停用词），BM25 算法认为这些词虽然重要性低，但并非毫无用处，可以通过参数 k_1 控制饱和度变化的速率和上限。k_1 值一般在 1.2~2.0，数值越低饱和的过程越快，在 Elasticsearch 中的默认取值为 1.2。

b 代表字段长度规约（Field-length Normalization），用于调整字段长度对相关性影响的大小，它可以将字段长度归约化到全部字段的平均长度上。BM25 算法认为较短字段比较长字段更重要，但字段中某个词的频度所带来的重要性会被这个字段长度抵消，因此需要考虑字段的平均长度。参数 b 的值在 0~1，1 代表全部归约化，0 代表不进行归约化。b 越大，字段长度对相关性的影响越大，反之越小，其在 Elasticsearch 中的默认取值为 0.75。

如果用 Q 表示输入的句子 Query，q_i 表示句子中的一个词，d 表示一个候选文档（字段），那么 BM25 算法的一般性公式为

$$\text{Score}(Q,d)=\sum_{i}^{n}W_i\cdot R(q_i,d)$$

其中 W_i 代表 q_i 的权重，通常用 IDF 表示，IDF 的计算公式为

$$\text{IDF}(q_i)=\log\frac{N-n(q_i)+0.5}{n(q_i)+0.5}$$

其中，N 代表全部文档数，$n(q_i)$ 代表包含 q_i 的文档数。

单词 q_i 与文档 d 的相关性得分 $R(q_i,d)$ 的计算公式为

$$R(q_i,d)=\frac{f_i\cdot(k_1+1)}{f_i+K}$$

$$K=k_1\cdot(1-b+b\cdot\frac{dl}{\text{avg}dl})$$

其中，dl 表示文档 d 的长度，avgdl 表示所有文档的平均长度。综上，BM25 算法的公式为

$$\text{Score}(Q,d)=\sum_{i}^{n}\text{IDF}(q_i)\cdot\frac{f_i\cdot(k_1+1)}{f_i+k_1\cdot(1-b+b\cdot\frac{dl}{\text{avg}dl})}$$

可见，Elasticsearch 中的 BM25 是一个词袋模型的算法，并没有考虑语义上的信息，例如“我喜欢你”和“我不喜欢你”在语义上是相反的，但基于 BM25 算法算出的相似度分值会非常高；而对于“你很漂亮”和“你很好看”，由于“漂亮”和“好看”是两个不同的词，计算出的相似度分值会较低。为了优化检索效果，我们通常会利用 Elasticsearch 对问答库进行粗筛选，再结合后续的匹配算法排序并选择候选答案。随着词嵌入方法的普及，匹配算法通常会先获取

候选句子的向量表示，这种向量的表示一定程度上包含了语义信息，进而通过向量之间的余弦距离计算出句子的语义相似度。

句向量的获取方法有很多种，下面将介绍几种经典的句子向量表示方法。由于句子向量通常由词向量通过监督方法或无监督方法获得，因此介绍句子向量之前，先介绍主流的词向量表示模型。传统的基于上下文共现关系的概率统计的词向量模型有 Word2vec、GloVe。随着深度学习的发展，有代表性的词向量有 FastText、ELMo、BERT 等。

由单词向量获取句向量的方法，最简单的有：向量加和平均、向量极值法等，但是这些方法并不能很好地获取句子的特征，而其他大多数方法属于监督学习模型，典型的有 Recursive Networks、Recurrent Networks、CNN、RCNN 等，主要针对有标签标注数据的分类任务展开，不具有一般性。Sanjeev Arora[5] 提出了一种无监督的句向量表示方法，对传统向量加和平均方法做了优化，将词频因素作为句子中每个单词向量的权重，并通过 PCA 或者 SVD 降维的方法移除句向量中的无关部分以获取核心内容，该方法只要提供一个大规模的文本用于统计词频即可，计算速度快，在多个数据集及多个 NLP 任务上取得了不输给 RNN 和 LSTM 的表现。后来，不断有研究者推出新的通用的句向量表示模型，其中具有代表性的有跳跃思维向量（Skip-Thought Vectors）[6]和快速思维向量（Quick-Thought Vectors）[7]。

Jamie Kiros 等人于 2015 年发表的跳跃思维向量，是一个通用无监督句子表示模型，该模型借鉴了 Word2vec 中 skip-gram 的模型。不同的是，Word2vec 中的 skip-gram 模型通过当前词预测上一个词和下一个词，而 Skip-Thought 则通过当前句子预测上一个句子和下一个句子。模型采用了端到端框架，将大量的包含上下文的纯文本数据作为训练数据集，其输入数据格式是一个包含上下文句子的三个句子的集合，并将得到的模型中 Encoder 部分作为特征抽取器（feature extractor），给任意句子生成向量。特别地，Skip-Thought 借鉴了 Tomas

Mikolov[8]解决机器翻译中缺失词的思路，提出了一种词汇表扩展的方法，将一个基于大规模数据集训练的 Word2vec 训练的词向量映射到 Skip-Thought 的词向量中，解决了未登录词（Out-Of-Vocabulary，OOV）的问题。

快速思维向量是跳跃思维向量的改进版，于 2018 年提出。其核心思想与跳跃思维向量相同，不同之处在于 Quick-Thought 的解码器由一个生成模型转变为一个分类模型，因此给定前一句话预测下一句话的任务被重新定义为一个分类任务，分类器需要在一组候选答案中选择一个合适的句子作为下一句的预测输出。从理论上解释，该模型将生成问题进行了区分性近似取值。这样做的最大好处是模型的训练速度比 Skip-Thought 快一个数量级，是一个非常好的在大规模数据集上进行训练的候选方案。

综上所述，基于检索的闲聊系统通过组合 Elasticsearch 搜索引擎和句向量模型，完成对输入问句的相似问句检索，并以返回问答库中最相似问句的答句作为输入问句的闲聊回复，整体流程如算法 5-1 所示。

算法 5-1　基于检索的闲聊系统实现流程

输入：问答库语料数据 P
大规模文本数据集 C
输入问句 S

过程

1．将全部对话语料 P 存入 Elasticsearch 数据库，自动生成索引。
2．利用大规模文本数据集训练词向量，进而选择任意一种句向量生成方法获取句向量模型。
3．输入句子 S，通过 Elasticsearch 搜索引擎获取候选相似问句集合 A。
4．由句向量模型分别为输入句子 S 和候选相似问句集合 A 生成句向量。
5．分别计算 S 与每一个 $a_i \in A$ 的余弦相似度，获得相似度分值 Score(S, a_i)。
6．根据 Score 对候选答案进行排序，选择最佳相似问句，并选择该问句的答句作为输入问句的闲聊回复。

输出：输入问句 S 的闲聊回复

5.3 基于生成的闲聊系统

5.3.1 基于生成的闲聊系统介绍

基于生成的闲聊系统较基于对话库检索的闲聊系统更复杂，其能够通过已有的语料生成新文本作为回答。基于生成的闲聊系统通常基于与机器翻译相关的技术，但是它并非把一种语言翻译成另一种语言，而是将输入文本“翻译”成输出文本（回答）。

生成式聊天机器人在接收到用户输入的句子后，采用一定的技术手段自动生成一句话作为应答，好处是可以覆盖任意话题的用户问句，缺点是生成式应答的句子质量很可能存在问题，比如语句不通顺、句法错误等看上去比较低级的错误。

生成式聊天机器人系统通过构建端到端的深度学习模型，从海量对话数据中自动学习“问题”和“回复”之间的语义关联，从而达到对任何用户问题都能自动生成回复的目的。需要注意的是，采用端到端的生成模型往往会出现安全回答的问题、机器人个性不一致的问题和多轮对话中的对话连续性问题。简单来说，生成式聊天机器人系统对于输入的句子，首先通过循环神经网络进行编码，然后通过解码输出对应回复句子中的每个词。可以看到，图 5-7 中，系统对输入句子进行编码，然后用编码指导词的输出，在输出时既考虑了原始句子的编码，也加上了不同层次的注意力机制，最后传递输出，直到输出词尾[9]。

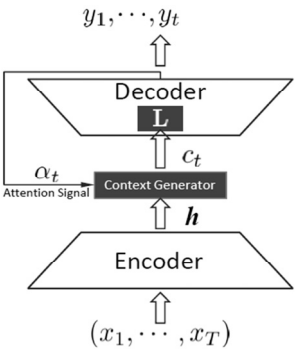

图 5-7 用于句子生成的编解码示意图

在实际操作中，生成句子时将所有词都看作等价的是不合适的，因为与输入句子或聊天主题关系更密切的词有较高的权重是符合人类认知的。参考基于对话库检索的闲聊系统在处理一条提问对应多条回答时使用的方法，为了在输出的概率方面体现出哪个词与主题的相关性更高，考虑将注意力模型嵌入编解码过程。另外，为了克服用传统的 RNN 及注意力模型建立的生成式闲聊系统中存在的回答过于枯燥的问题，在实际操作中往往要用到外部知识来丰富回答。一种常用的基于外部知识提高回复多样性的方法是主题词增义，即在使用一般的端到端方法预测回复的单词序列的同时，通过增强与输入句子有关的主题词，对主题词进行编码，预测输出的单词序列。通过应用上述方法，除了来自源端的信息，回复的词还受到了主题词的制约。

微软发布的聊天机器人小冰在一定程度上采用了生成式回复技术，其框架如图 5-8 所示。

通过 RNN 将输入句子编码成向量，同时利用话题模型引入话题信息，在解码的过程中，利用编码向量和话题信息生成对应的回复。通过引入注意力机制，输入的每个向量和话题关键词都被进行加权处理，以使重要信息对回复产生更大的影响。

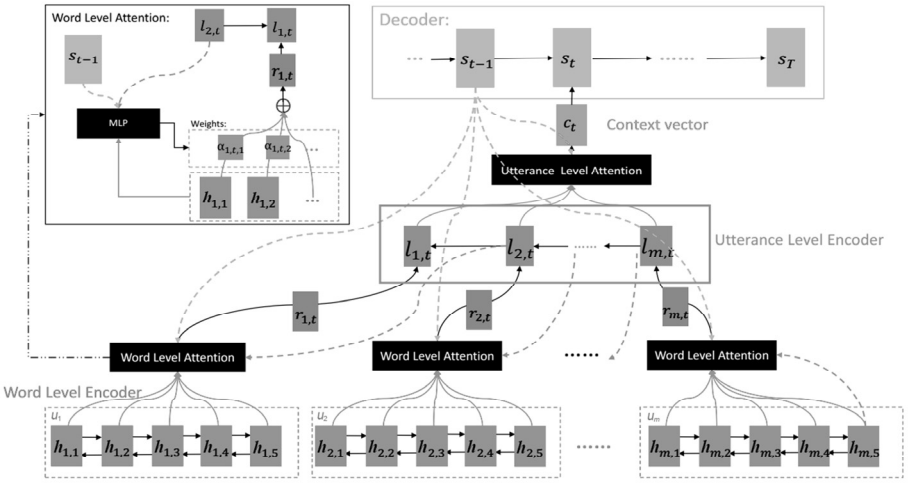

图 5-8　微软小冰的技术框架

5.3.2　生成式闲聊系统的新发展

为了克服生成式闲聊系统的缺陷，改进生成式闲聊系统的效果，研究人员在安全回答、个性一致、对话连续性 3 个方向上进行了改进研究。

Cho 等人[10]是较早提出 Encoder-Decoder 模型的研究人员，这一模型最初被用于机器翻译中，算法的大致思路为：每次向 Encoder 的 RNN cell 输入一个词的词向量，Encoder 对输入的词向量依次进行编码，直至得到整个句子的语义向量表示，然后由 Decoder 根据句子的语义向量表示输出目标回复。

2014 年，对生成式聊天机器人具有深远影响的 seq2seq 发布了。它克服了 RNN 无法完成端到端映射的缺陷。Sutskever 等人[11]通过 LSTM 方法将输入的序列映射为固定长度的向量，然后使用深度 LSTM 从已知的向量中解码得到目标输出序列。Sutskever 等人将 seq2seq 应用于英语和法语之间的机器翻译中，并使用 BLUE 方法对模型的性能进行检验。下面笔者结合论文中的示意图对 seq2seq 方法的大致思路进行介绍。

假设输入序列是“ABC”，目标输出序列是“WXYZ”，<EOS>是 end of Sentence 的缩写，Encoder LSTM 每次读取一个单词的词向量，且对所有已输入的词向量进行编码，当“ABC”均被输入后，Encoder LSTM 将生成可以表示“ABC”序列的一定长度的向量 $\boldsymbol{c}$。然后，Decoder LSTM 根据向量 $\boldsymbol{c}$，每次预测出一个下一时刻的词汇。如图 5-9 所示，输入“ABC”的结束标志后，预测出“W”，然后依次预测出“X”“Y”“Z”和结束标志。

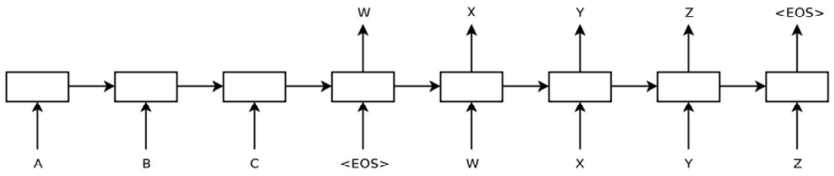

图 5-9 seq2seq 处理示意图

上述 seq2seq 方法在实际使用时有 3 个主要的缺陷。

（1）难以保持机器人个性的一致性，在用户输入“你几岁了”和“你今年多大了”时，机器人往往会给出不同的答案。

（2）机器人往往会给出“哈哈”“呵呵”等无意义的安全回答。出现安全回答是因为训练数据中（人与人实际交往时）对这些词的使用较为频繁，导致机器学习的过程中这类回答较容易被抽取。

（3）长对话语义的保存或机器人的记忆问题，如代词指代较远位置的上文中的内容，机器人难以像人脑一样顺利进行指代消解。

笔者在叙述聊天机器人与深度学习相关的内容时，对上述 3 个问题出现的原因、导致的结果都已进行了初步阐述，下面分别对这 3 个问题的研究进展进行叙述。

2016 年，斯坦福大学和微软研究院的研究人员共同发布了基于个性的神经

聊天机器人模型[12]，通过在 Encoder 时增加用户个性信息（用户向量，在生成词嵌入的过程中同时生成）和用户回复信息（预测用户回复另一用户时的回复，实际上是通过对用户 *i* 的向量和用户 *j* 的向量做线性组合变换对用户 *i* 回复用户 *j* 的行为进行建模），解决聊天过程中机器人个性一致性的问题。2017 年，李纪为博士等人[13]发表了名为 *Adversarial Learning for Neural Dialogue Generation* 的论文。在这篇论文中，李纪为博士等人将回复生成问题作为强化学习问题，使用对抗生成的思想训练生成模型。Serban 等人[14]将端到端的分层 RNN 思想引入开放领域的回复生成问题，同时在句子层面和对话语境层面进行建模，通过建立长对话向量帮助机器记忆长对话的语义。

因为人在说话的时候是考虑上下文的，不是只看当前的一句话，生成模型需要将多轮对话的信息都考虑进去，所以，对 session 进行编码，用 session 预测输出的回复，在多轮对话中有显著意义。例如，可以使用多层感知的方法模拟多轮对话。这种多层感知的方法对之前出现的所有句子分别进行编码，每个编码都可以体现整个句子的信息，再通过注意力模型与目标连接，预测时通过基于句子的注意力模型对回复进行预测。

基于生成的闲聊系统在避免安全回答、个性一致和上下文建模 3 个方向具有较大的改进空间和可能性。

在上下文建模方面，Xing 等人的研究[15]给出了一种解决多轮对话上下文建模的思路和方案，通过构建多层的注意力框架，同时提取词向量和句向量，以确保所有上下文有效信息均被提取，进而提升闲聊系统在多轮对话中的整体表现。

上面提到的聊天机器人表现出来的安全回答问题、个性一致性问题和上下文连贯性问题，是目前行业公认的聊天机器人存在的主要问题，这些问题带来了不良的用户体验。Facebook 的研究人员认为，如果有一个质量较高的、公开

的聊天数据集，上面的一些问题就能得到比较好的解决。因此，Facebook 的工程师建立了聊天数据集 Persona-Chat 用以训练聊天机器人。Persona-Chat 数据集包含了逾 16 万条对话。

5.3.3 基于生成的闲聊系统实现

本节，我们借助 TensorFlow 提供的 seq2seq[①]模型，实现基于生成的闲聊系统。为了阐述方便，在 TensorFlow 的 seq2seq 模型中，词嵌入采用的是 one-hot 编码方式，即每个词由一个独立的数字编码代替，这种词的表示方式虽然会影响模型效果，但是在演示中会更加直观。

在使用 TensorFlow 的 seq2seq 模型时，先引入以下依赖。

```
import tensorflow as tf
from tensorflow.models.rnn.translate import seq2seq_model
```

seq2seq 模型位于“tensorflow/tensorflow/python/ops/seq2seq.py”目录下。

如算法 5-2 所示，模型先对训练语料进行文本预处理，包括典型的分词、去停用词等。同时，为了提升泛化能力、增强模型效果，可以将一些实体进行通用的标识，例如将人名替换为“name”、将地名替换为“location”等。然后，建立词表，并以 one-hot 编码方式对词进行编码，例如将“喜欢”映射为“67”、将“电视”映射为“88”。由此，各个语句就转化成了数字编码所组成的向量形式，如图 5-10 所示。到此为止，整体的数据预处理工作结束，训练数据可以被 TensorFlow 的 seq2seq 模型所接受。下一步便进行模型训练，神经网络通过输入的问答对自动调整参数，生成问题–回答模型。在模型验证阶段，输入测试语句，验证模型输出是否正确，演示结果如图 5-11 所示。

① https://www.tensorflow.org/tutorials/seq2seq

算法 5-2　利用 TensorFlow 提供的 seq2seq 模型实现生成式闲聊对话

输入：训练语料集

过程：

1. 训练数据预处理，包括基本的分词、去停用词等。
2. 语句转特征向量，以 one-hot 编码方式对词汇进行映射。
3. 输入数据，训练模型。
4. 模型验证。

输出：训练完毕的 seq2seq 模型

```
18 52 30
13 748 10 54 18 63 688 76 145 380
14 28 111 54 53 110 20 544 664 38
23
2869 793
23 362 23 459
4 209 6 459 13 111
375 291 1002 29
495 791 495 791 22 22
2622 2238 52 182
405 228 4 23
13 53 84 63 148 57
4 112 43 205 36 82
405 729 64 17 88
```

图 5-10　one-hot 编码示例

图 5-11　基于生成的闲聊系统实现演示结果

5.4 参考文献

[1] Huang P S , He X , Gao J , et al. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data.Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management. ACM, 2013.

[2] Hu B , Lu Z , Li H , et al. Convolutional Neural Network Architectures for Matching Natural Language Sentences. 2015.

[3] Pang L , Lan Y , Guo J , et al. Text Matching as Image Recognition. 2016.

[4] Lowe R, Pow N, Serban I, et al. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. Computer Science, 2015.

[5] Arora S, Liang Y, Ma T. A Simple But Tough-to-beat Baseline for Sentence Embeddings. 2016.

[6] Kiros R, Zhu Y, Salakhutdinov R R, et al. Skip-thought Vectors.Advances in Neural Information Processing Systems. 2015: 3294-3302.

[7] Logeswaran L, Lee H. An Efficient Framework for Learning Sentence Representations. arXiv preprint arXiv:1803.02893, 2018.

[8] Mikolov T, Le Q V, Sutskever I. Exploiting Similarities among Languages for Machine Translation. arXiv preprint arXiv:1309.4168, 2013.

[9] L. Shang, et al, Neural Responding Machine for Short-Text Conversation, ACL 2015.

[10] K. Cho, B. Van Merrienboer, C. Gulcehre, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, Computer Science, 2014.

[11] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to Sequence Learning with Neural Networks, vol. 4, pp. 3104-3112, 2014.

[12] J. Li, M. Galley, C. Brockett, et al. A Persona-Based Neural Conversation Model, 2016.

[13] J. Li, W. Monroe, T. Shi, et al. Adversarial Learning for Neural Dialogue Generation, 2017.

[14] I. V. Serban, A. Sordoni, Y. Bengio, et al. Building End-to-end Dialogue Systems Using Generative Hierarchical Neural Network Models. 2015.

[15] Xing C, Wu W, Wu Y, et al. Hierarchical Recurrent Attention Network for Response Generation. 2017.

6 聊天机器人系统评测

6.1 问答系统评测

客观而科学地评测问答系统、对话系统和闲聊系统的性能，是评判一个聊天机器人智能程度的关键问题之一，而评测数据集既是衡量和评估聊天机器人系统性能的基础，也是很多商业系统取胜的法宝。一般来说，出于商业竞争和隐私保护的原因，标注后的真实数据和生成的评测数据集不会轻易公开。对聊天机器人从业者来说，高质量对话数据集缺乏是普遍难题。

目前，针对问答系统，公开的评测会议主要有 TREC QA Track、NTCIR 的 QA 评测、QALD 评测、INEX Linked Data 评测、Semantic Search 挑战、Bio ASQ、EPCQA 等，针对对话系统的最有影响力的评测是由微软公司发起的 DSTC 评测。6.1 节和 6.2 节将分别对问答系统评测和对话系统评测进行介绍。

6.1.1 问答系统评测会议

1. TREC QA Track

1999 年，文本检索会议（Text REtrieval Conference，TREC）[①]引入了问答系统专项评测（Question Answering Track，QA Track）：TREC QA Track 是针对英文问答的评测平台，从 1999 年至 2018 年共开展了 13 届，主要针对事实性问题进行问答，每年会议会提供包含几百个问题和答案的数据集对参赛系统进行评测。TREC 的主要组成部分如图 6-1 所示。

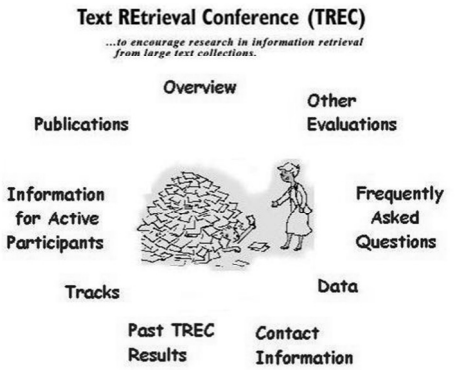

图 6-1　TREC 的主要组成部分

虽然每年的任务都有变化，但总体来说主要有以下几类。

（1）Factoid 类：测试系统对基于事实、有简短答案的提问的处理能力，例如伯利兹城坐落在哪里（Where is Belize located），但不包含需要总结、概括类的问题，例如如何办理出国手续。

（2）List 类：要求系统列出满足条件的几个答案。TREC 2003 要求系统尽可能多地给出满足条件的实例，例如手机制造商列表（List the names of cell phone manufacturers）。

① https://trec.nist.gov

（3）Definition 类：要求系统给出某个概念、术语或现象的定义和解释。例如宝莱坞是什么（What is Bollywood）。

（4）Context 类：测试系统对相关联的系列提问的处理能力，即对提问 *i* 的回答依赖于对提问 *j*（*i* 依赖于 *j*）的理解。例如：

a. 佛罗伦萨的哪家博物馆在 1993 年遭到炸弹的摧毁?

b. 这次爆炸发生在哪一天?

c. 有多少人在这次爆炸中受伤?

（5）Passage 类：从 TREC 2003 开始提出的任务。和其他任务不同的是，这类任务对答案的要求偏低，不需要系统给出精确答案，只要给出包含答案的一个字符序列（a small chunk of text that contains an answer）。

（6）Other 类：TREC 2004 定义的任务。TREC 2004 的测试集包括 65 个目标，每个目标由数个 Factoid 问题、0～2 个 List 问题和 1 个 Other 问题组成。其中，Other 问题的返回答案应该是一个非空的、无序的、无限定的关于这个目标的描述，且不包括 Factoid、List 问题已经回答的内容。

TREC QA Track 的主要评测指标有平均排序倒数（Mean Reciprocal Rank，MRR）、正确率（Accuracy）、加权置信度分值（Confidence Weighted Score，CWS）等。

从 2015 年开始，TREC QA Track 注重问题的实时性，参加比赛的系统将回答最新的、最真实的用户问题。这些问题来自 Yahoo!Answers 上用户的真实提问，若某个提问没有及时被其他网民回答，则这一问题会被自动分配给参赛系统来回答。从 2015 年到 2018 年，TREC QA Track Live 已经开展了 4 次。

TREC LiveQA 2015[①]的任务主题有 Arts&Humanities、Beauty&Style、Computers&Internet、Health、Home&Garden、Pets、Sports、Travel 等。TREC LiveQA 2016[②]的任务主题与 TREC LiveQA 2015 基本相同，除了去掉评估难度过大的 Computer&Internet 类问题。TREC LiveQA 2017[③]的主任务还是来自 TREC LiveQA 2015 和 TREC LiveQA 2016，数据集也采用前两年的，主要的变化是添加了专门针对医疗的子任务（medical subtask）。

TREC QA Track 的贡献主要有以下两点。

（1）TREC QA Track 每年都会提供 500 道左右的测试问题，经过将近 10 年的评测，建立起了含有数千道问题的题库。这些问题、答案、答案模板和证据，组成了此后自动问答研究的标准语料库，极大地促进了自动问答的研究水平。

（2）TREC QA Track 评测的另一项贡献是提出了适用于 QA 的评价指标。TREC QA Track 提出的第一种指标是正确率（指的是回答正确的问题占问题总数的百分比）。在系统仅为每个问题提供一个答案时，可用这一指标进行评测（2003 年和 2004 年的问答评测都使用了该指标）。2007 年的问答评测则采用了正确率的一种变体，即将答案是否正确进一步细化为全局正确、局部正确（文档集中存在该答案，但该答案并非是整个文档集中的最佳答案）、不确切（与正确答案有交集）、不正确、不支持（答案正确，但给出的证据不支持答案）这 5 种结果，并为每种结果设置不同的权重。

2. NTCIR 的问答评测

NTCIR 的全称是 NII Testbeds and Community for Information access

① https://sites.google.com/site/trecliveqa2015/
② https://sites.google.com/site/trecliveqa2016/
③ https://sites.google.com/site/trecliveqa2017/

Research。NTCIR Workshop[①]是一系列专门为提升信息获取技术（Information Access Technology）而设计的评测研讨会，包含信息检索、问答、文本摘要提取（Text Summarization Extraction）等任务。从 NTCIR-3 2001 年开始，NTCIR 加入了问答评测的内容，问答的范围主要来自日本的《每日新闻》（*Mainichi Newspaper*）中的文章。日语问答评测平台 QAC（Question Answering Challenge）从 2002 年开始，QAC-1（NTCIR-3）定义的 3 个子任务如下。

任务 1：共 100 个问题，系统为每个问题给出 5 个按概率大小排列的答案列表；采用 MRR 打分标准；系统必须给出支持每个答案的全部文档。

任务 2：共 100 个问题，每个问题只能有一个答案，且系统必须给出支持该答案的全部文档。

任务 3：这个任务评测系统对关联问题的处理能力；关联问题是指问题之间可能有互指关系、省略等，类似于 TREC 中的 Context Task；系统必须给出支持每个答案的文档。

从 NTCIR-3 到 NTCIR-6 连续开展 4 期 QAC 任务，每期的任务类别基本相同，仅评测问题的侧重点有所不同。

NTCIR-7 和 NTCIR-8，从原来仅针对事实类的问答评测转向跨语言的信息获取，即 Advanced Cross-Lingual Information Access（ACLIA）。IR 往往是 QA 任务中不可分割的重要方法之一，因此，将 QA 和 IR 划归到一个大任务下，分为 QA 和 IR 两个方向的子任务。

1）复杂跨语言问答子任务［Complex Cross-lingual Question Answering（CCLQA）subtask］

前 4 期 QAC 问答中只针对较简单的事实性问答。为了进一步丰富问答评

① http://research.nii.ac.jp/ntcir/workshop/index.html

测，CCLQA 将融入更复杂的问答，同时实现跨语言的问答，以及多语言的答案融合。

2）信息检索问答子任务［IR for QA（IR4QA）subtask］

是对先前 CLIR（Cross-Lingual Information Retrieval Task）任务的发展，以 XML 形式对输入输出进行规范，因此可以将原来的 CLIR 组件融合到 CCLQA 评测中。同时，可以根据 CCLQA 评测结果的好坏，衡量 CLIR 组件自身的好坏。

NTCIR-9 和 NTCIR-10 针对问答的评测又发生了变化，文本中的推理识别［Recognizing Inference in TExt（RITE）］任务，作为一个在各种 NLP /信息访问研究领域（例如信息检索、问答、文本摘要）中处理常见语义处理的通用基准任务，分为二分类子任务［Binary-Class（BC）subtask］、多分类子任务［Multi-Class（MC）subtask］、入学考试子任务［Entrance Exam subtask（Japanese Only）］、文本中的推理识别子任务（RITE4QA subtask）4 个子任务。

从 NTCIR-11（2013—2014 年）开始，NTCIR 将问答发展到 QA Lab，QA Lab 的目标是提供一个基于模块的平台，用于系统性能评估和比较，以解决更为实际的问题：大学入学考试问题。QA Lab 提供的问答系统框架分为 4 个部分（如图 6-2 所示）：问题分析、文档检索（Document Retrieval）、抽取候选答案（Extracting Answer Candidates）和答案生成（Answer Generation）。

不同于 TREC QA Track，直至最新的 NTCIR-14（2018—2019 年），NTCIR 每次的任务题型大致相同[①]，主要分为以下 3 类：

（1）多选题（multiple-choice questions）。

① http://research.nii.ac.jp/qalab/task.html

（2）术语问题（term questions）。

（3）问答题（essay questions）。

NTCIR 的问答系统评测框架如图 6-3 所示。

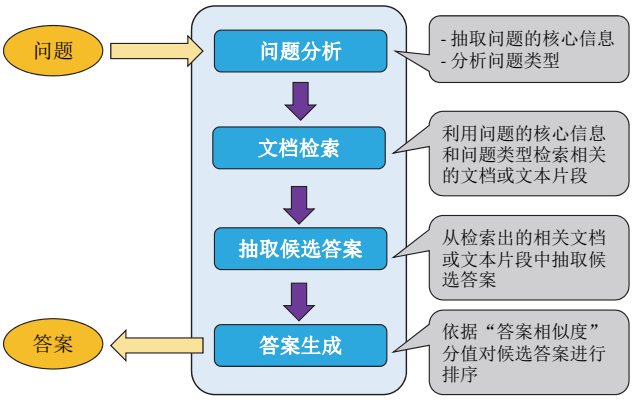

图 6-2　QA Lab 的示意图

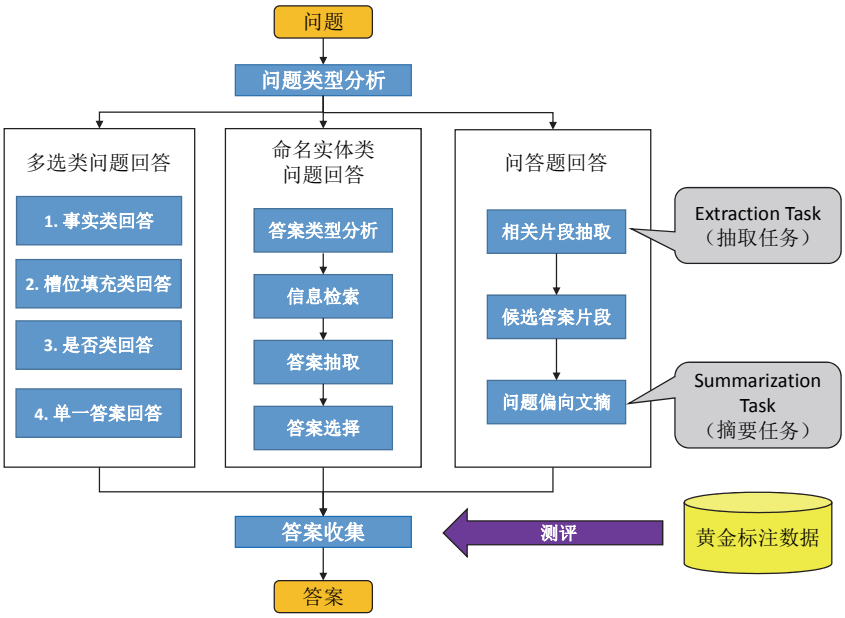

图 6-3　NTCIR 的问答系统评测框架

如图 6-3 所示，输入问题，通过问题类型分析模块确定问题的类型，然后分别进入不同的模块进行问题答案的搜集；由于问答题比较复杂并且难于评测，NTCIR 又增加了抽取任务和摘要任务两个子任务模块。参赛者通过黄金问答数据对训练问答系统，并进行评测和优化，最终获得问题所对应的答案。

3. CLEF 的 QALD 评测

CLEF 评测的一般场景是 CLEF QA Track（CLEF Question Answering Track），主要针对两类任务：一类是（生物医学）医学专家任务，另一类是针对开放领域（QALD 和入学考试）的任务。

这里我们仅着重介绍与聊天机器人问答系统密切相关的 QALD 针对 KBQA[①]的评测。QALD 是一系列多语链接数据问答系统的评测竞赛活动，是 CLEF（CLEF Question Answering Track）中针对 KBQA 系统的问答评测任务。QALD 评测框架如图 6-4 所示。

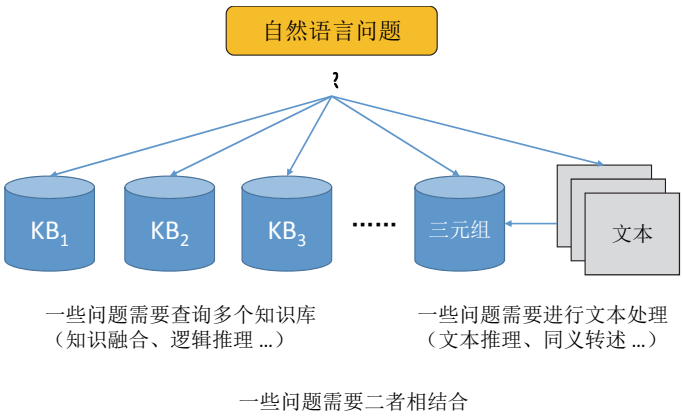

图 6-4　QALD 评测框架示意

QALD 的主要目的是提供一个统一的评测基准，以深入分析语义问答系统的优缺点。QALD 评测系统的输入可以是自然语言或 RDF 格式的数据，根据数

① http://nlp.uned.es/clef-qa/

据集和相关的知识源及自然语言的问题或关键字，问答系统直接返回一个正确答案或者一个用于检索答案的 SPARQL 查询语句。

QALD 从 2011 年开始到目前共举办了 8 次竞赛，从 QALD-1 到 QALD-7 均是在 ESWC（Multilinguality、Semantic Web、Human-Machine-Interfaces）会议上举行的。从 QALD-8 到即将开始的 QALD-9，则在 ISWC（International Semantic Web Conference）会议的 NLIWoD（Natural Language Interfaces for Web of Data）研讨会上举行。

QALD 竞赛的主要任务有如下 3 类。

- 多语问答，基于 DBpedia
- 混合问答，基于 RDF 和非结构化的自由文本数据
- 问答基于 RDF 多维度数据集（data cubes）的统计数据

为了建立一个统一的基准，QALD 由以下基本成分构成。

（1）数据集：一个 RDF 格式的链接数据的资源。数据的主要来源是 DBpedia[①]、Yago[②]和 MusicBrainz[③]。

（2）Gold standard：一些自然语言问题被标注成对应的 SPARQL 序列和答案以供训练和评测。图 6-5 展示了一个 XML 格式的训练集。

（3）评测方法：包括一系列程序和指标，用来衡量参赛系统的表现。QALD 的评测指标包括召回率、精确率和 F-测度。

（4）基本构成：由一个 SPARQL 端点（SPARQL endpoint）和一个在线评

① http://dbpedia.org
② http://www.mpi-inf.mpg.de/yago-naga/yago/
③ https://musicbrainz.org

测工具构成，可以评估参赛系统反馈答案的正确性。

```
<question id="36" answertype="resource" aggregation="false" onlydbo="false">
  <string>Through which countries does the Yenisei river flow?</string>
  <keywords>Yenisei river, flow through, country</keywords>
  <query>
   PREFIX res:  <http://dbpedia.org/resource/>
   PREFIX dbp:  <http://dbpedia.org/property/>
   SELECT DISTINCT ?uri ?string WHERE {
        res:Yenisei_River dbp:country ?uri .
        OPTIONAL {?uri rdfs:label ?string . FILTER (lang(?string) = "en") }
   }
   </query>
  <answers>
    <answer>
     <uri>http://dbpedia.org/resource/Mongolia</uri>
     <string>Mongolia</string>
    </answer>
    <answer>
     <uri>http://dbpedia.org/resource/Russia</uri>
     <string>Russia</string>
    </answer>
  </answers>
</question>
```

图 6-5　XML 格式的训练集示例

4．INEX Linked Data Track

INEX Linked Data Track 在 2011—2013 年每年举办一次，共开展了 3 届，其评测的重点是结合文本和结构化的数据。

INEX Linked Data Track 的数据集

- 英文维基百科（MediaWiki XML Format）
- DBpedia 3.8 & Yago2（RDF）

INEX Linked Data Track 的评测任务

- Ad-hoc 任务：对于关键字查询格式的查询请求，返回一个排序后的结果列表作为对查询请求的响应（共包括 144 个查询主题）。
- Jeopardy 任务：基于一组自然语言 Jeopardy 线索检测检索技术的效果（2012 年的任务包括 74 个检索主题，2013 年的任务包括 31 个检索

主题）。

5．Semantic Search 挑战[①]

Semantic Search 挑战的重点在于关联数据集上的实体链接与检索。

- 数据集为从公开来源抽取的三元组链接数据。
- 任务主要包括实体检索和列表检索。实体检索指查询一个特定的实体，例如问洛杉矶加州、IBM、纽约的邮政编码等问题；列表检索指查询与特定标准相匹配的对象，查询的条件由组委会人员手工编写，例如，列举 10 个古希腊城市、列举以葡萄牙语为官方语言的国家等。

6. Bio ASQ Workshop

Bio ASQ Workshop 的数据集

- PubMed 文档

Bio ASQ Workshop 的 3 个任务

- **大规模在线生物医学领域语义索引**

在这个任务中，参赛者需要对新的未标注的公开医学文档进行标注和分类，任务将对标注结果的表现进行评测。

- **介绍性的生物医学领域语义问答**

这个任务利用基准数据集中包含的开发和测试问题，以及最标准的答案，构成了一个医学专家系统。参赛者需要从制定的资源中找到答案，并反馈相关的概念、文章、段落及 RDF 三元组。

① http://semsearch.yahoo.com/datasets.php#

- **基于生物医学文献的信息提取**

在这个任务中，参赛者的系统需要从 PubMed Central 提供的文件中提取全部的候选序号和候选内容，由 PubMed Central 的参考标注结果来评测参赛系统的信息抽取表现。

7. EPCQA——汉语问答评测[1]

汉语问答系统的评测起步较晚，一直没有一个公认的评测系统及评估方法。作为尝试，中科院自动化所建立了一个汉语问答系统评测平台（EPCQA），其语料库、测试集和打分标准参考 TREC QA Track 和 CLEF 的成功经验，并结合汉语的特点做了适当的调整。

EPCQA 的语料库大小约为 1.8GB，内容主要来自互联网网页，涉及国内外、娱乐、体育、社会和财经等领域。EPCQA 测试集的建立遵循全面性、真实性和无歧义性 3 个原则，包含 4250 个涉及事实、列表和描述类问题，数据来源的渠道主要有自然语言搜索网站日志、百科知识问答题库、实验室工作人员、英语提问的翻译等。其中，疑问词的提问、表达模糊的提问、回答内容为完成某件事的过程而非简短答案的提问这些类型的问题，不作为考察的范围。

百科知识问答题库中的问题描述得都很书面化，不能直观地反映用户真实提问的场景。因此，EPCQA 对问题进行了一些口语化的处理。例如，将百科知识问答题库中的提问“空气的主要组成成分是氮气、氧气、二氧化碳和其他稀有气体，其中氧气所占的百分比是多少”处理成“氧气占空气的比重是多少”。

同时，对英语问题库的“翻译”也是获取汉语问答系统测试集的另一个非常重要的途径。其中，英语问句的来源主要是历届 TREC 比赛的测试集。这里的“翻译”并不全是对英语问题的直译，还需要对部分可能在中文中找不出答案的问题在不改变问题类型的情况下做适当的修改，例如：

英语问题：Who wrote “East is east, west is west and never the twain shall meet” ?

对应的中文问题：名著《红楼梦》是谁的作品?

英语问题：What is the name of CEO of Apricot Computer ?

对应的中文问题：联想公司的 CEO 叫什么名字?

EPCQA 针对不同类型的问题采用不同的打分标准。EPCQA 初步拟定，事实提问采用 MRR 打分标准，参考 TREC。列表提问采用实例召回率（Instance Recall）、实例精确率（Instance Precision）、F-Measure（F1）等打分准则，参考 CLEF。对每一个问题，评测系统会列出一个基本信息和可接受信息的列表。基本信息是指该问题的答案中必须包含的部分；可接受信息是指可以构成一个正确答案的可选内容，但不是必需的信息。答案中超出基本信息和可接受信息的部分会在评分体系中减分。EPCQA 还未发展成熟，它设想的发展阶段为命名实体阶段、组块阶段、句群阶段和摘要阶段。

6.1.2 问答系统评测数据集

TREC QA Track Data[①]数据集

- 每届会议包括几百个问题的答案的数据集。
- TERC 1999 ~ TERC 2007 的数据来源主要是 FAQ Finder 的系统日志、TIPSTER 和 TREC disks 的报纸新闻文章和 AQUAINT disks。
- TERC 2015 ~ TREC 2017 的问题来自 Yahoo Answers 上用户问出的真实问题。
- TERC-8 的 200 个问题主要来自 NIST assessors 及 FAQ Finder 的系统

① http://trec.nist.gov/data/qamain.html

日志文件。

- TREC-9 (2000)的数据主要来自 TIPSTER 和 TREC disks 的报纸新闻文章，如 *AP newswire* (Disks 1-3)、*The Wall Street Journal* (Disks 1-2)、*San Jose Mercury News* (Disk 3)、*Financial Times* (Disk 4)、*Los Angeles Times* (Disk 5)、Foreign Broadcast Information Service (FBIS) (Disk 5)等。
- TREC-10(2001)的数据与 TREC-9 相同。
- TREC-10(2002)的数据来自 AQUAINT disks 的文档，主要问题来自微软 MSNSearch logs 和 AskJeeves logs。
- TREC-11(2003)、TREC-12(2004)的数据与 TREC-10(2002)的数据相同。

Free917

- 主要使用来源于 Freebase 的数据，共包含 917 个标注了逻辑表达式的问题。

WebQuestions

- 主要使用来源于 Freebase 的数据，包含 5810 组问题对，词汇表包含了 4525 个词，问题主要通过 Google Suggest API 爬取。
- 利用 Amazon Mechanical Turk 服务得到答案，特别是一个问题可能存在多个答案的时候。
- WebQuestions 提供了每个答案对应在知识库中的主题节点(topic node)。
- 利用 Average F1 对回答进行评价。

QALD

- 基于知识库的问答评测，主要使用 DBpedia 的数据，每年包含约 100 个问题。
- Hybrid Track：需要结合结构化数据和纯文本数据生成答案，必须依

靠文本信息。

Simple Questions

- 包含 108442 个简单的问题，每个问题附带一条 Freebase 三元组作为答案。

SQuAD[①]

- 阅读理解类问题，主要采用维基百科的数据。SQuAD 2.0 包含 150000 个问题，问题需要综合理解文本段落后得出，通常不能直接回答。

6.1.3 问答系统评测标准

问答系统评测常用的标准包括召回率、精确率和 F-测度。我们通过以下假设说明上述 3 个标准的计算方式和意义。

假设原始样本中有两类，其中共有 P 个类别为 1 的样本，以及 N 个类别为 0 的样本。经过分类，有 TP 个类别为 1 的样本被系统正确判定为类别 1，有 FN 个类别为 1 的样本被系统误判定为类别 0，有 FP 个类别为 0 的样本被系统误判定为类别 1，有 N 个类别为 0 的样本被系统正确判定为类别 0。

基于上述假设可以知道：P=TP+FN，N=FP+TN。

由此可以定义评测标准：

精确率，反映了被分类器判定的正例中真正的正例样本的比重。

$$P = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FN}}$$

准确率，反映了分类系统对整个样本的判定能力。

① https://rajpurkar.github.io/SQuAD-explorer/

$$A=\frac{\mathrm{TP}+\mathrm{TN}}{P+N}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}}$$

召回率，反映了被正确判定的正例占总的正例的比重。

$$R=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}=1-\frac{\mathrm{FN}}{P}$$

F-测度（F-Measure or balanced F-Score），就是通常所说的 F1 值。

$$F\text{-Measure}=\frac{2\times P\times R}{P+R}$$

6.2 对话系统评测

随着任务型对话系统的诞生，与之对应的评测方法也逐渐成为一个活跃的研究方向。任务型对话系统由任务驱动，通常涉及多轮次的对话场景，可以看作一个决策的过程，对任务驱动的多轮对话系统的评估通常是通过评估整体对话系统的效果实现的。任务驱动的多轮对话系统的核心目的是帮助用户有效地获取信息或服务，因此评估任务型对话系统最直接的两个指标是**对话成功率**和**对话成本消耗**（如对话时长、系统给出确认性回复所需的对话轮数等）。随后，研究人员在实际任务中发现对话成功率和对话长度是衡量对话系统优劣最重要的两个指标，后来的研究趋势也转为最大化对话成功率与最小化对话长度，并以此作为任务型对话系统评测的指标。然而，对话系统在与人进行实际交互时，任务完成的程度很难界定。主要的评测方法有 3 种，分别是数据驱动型对话评价模型方法、用户模拟评价方法和人工评价方法。

首先被广泛讨论和研究的方向是基于标注语料的数据驱动型对话评价模型。它离不开优质的训练数据，其中涉及的算法如协同过滤、重塑反馈函数等也证明了这一点。优质的训练数据对于对话系统的生成结果至关重要。但是，优质

的标注数据非常难获得，有研究者提出用机器模拟人类标注数据的过程来标注数据，也有研究者提出通过采用多种方式相结合的办法对数据进行自动标注的主动学习（active learning）标注。

尽管用于评价数据驱动的自然语言处理任务的方法有很多，但是由于对话系统本身具有的多轮交互的特性，导致评价对话系统的难度要大于评价一般的有明确评价指标的自然语言处理任务，如语言模型（language model）的评价指标混淆度（perplexity）、机器翻译中的 BLEU 值，以及自动文本摘要中的 ROUGE 值等。尽管目前已经有很多针对不同指标的评价矩阵，但如何综合利用这些评价矩阵来评价对话系统仍然是一个难以攻克的难题。事实上，对话系统评价的最终目标是评测用户的满意度，但总有许多因素导致评价结果与用户的真实感受和体验无法完全吻合，即使通过人工制定的多项指标对系统进行评测，也会有不同程度的偏差，并且难以罗列出所有的特征并加以对比，从而达到全面的评价效果。因此，现存的数据驱动的评价过程和评价方法大都无法准确地满足用户的要求。

对于对话系统，用户模拟评价是最有效、最简单的评价策略，通过模拟不同情境下的对话，可以尽可能地覆盖最大的对话空间，并且能够在大范围场景下进行有效的测试和评价。然而，这种方法的缺点也很明显，那就是真实用户的反应与用户模拟器的反应之间必然存在差异，这个差异对评价结果准确性影响的大小主要取决于用户模拟器的好坏，用户模拟器很难完全模拟人的真实反应。即使存在上述缺点，用户模拟仍然是评价任务型对话系统最常用的方法。

人工评价是指通过雇佣专门的评测人员对对话系统生成的结果进行评价，这样做的好处显而易见，最符合人的真实感受和体验，并且能够产生更多真实的评价数据。目前，这种评价方法主要出现在实验室等研究资源雄厚的环境中，评测人员在特定的任务领域内对系统进行评测，根据预设的各种询问方式与系统进行对话，依据对话的效果对系统的表现进行评分。人为的评分必然带有一

定主见性，不论雇佣的评测人员的评价是否能够全方位地代表用户的真实感受。这种方法最大的问题在于如何雇佣足够多的评测人员（很明显，这需要大量的开销），后期衍生出的众包模式及借助网络媒介在网络上进行实时评价等方法都可以一定程度上解决该问题。除了开销巨大，这种方法还存在众包选择的评测人员是否专业，评判标准是否统一，能否真实代表所有用户的问题。事实上，如果没有很好地监控人工标注质量，人工评价的结果将直接影响对话系统的评测结果。也有事实证明，人工评价很多时候并没有非常准确地表现出对话的准确程度。

6.2.1 对话系统评测会议

对话系统领域最具影响力的评测会议便是由微软公司发起的 DSTC 评测会议。在对话系统中，状态追踪（state tracking）是指准确地估计用户的目标促进对话的进展。精确的状态追踪是非常重要的，因为它可以有效减少语音识别中的错误，并且有助于减少在诸如对话这样的过程中固有的模糊性。

DSTC 评测会议从 2013 年开展第一届，截至 2018 年共开展了 6 届，主要针对一些现实的应用场景进行对话评测，如公交车路线咨询、餐馆咨询、旅游咨询等。图 6-6 给出了餐馆信息领域的 8 个不同槽位和槽位说明，比如食物（food）包括 91 个可能的值，并且支持用户通过食物名称搜索餐馆。举例来说，用户的问题可以是“我想找一个能吃意大利菜的餐馆”，其“食物”的槽位信息是“意大利菜”。同理，图 6-7 给出了 DSTC3 的槽位说明。

Slot	User may give as a constraint?
area	Yes, 5 possible values
food	Yes, 91 possible values
name	Yes, 113 possible values
pricerange	Yes, 3 possible values
addr	No
phone	No
postcode	No
signature	No

图 6-6　餐馆信息领域的对话槽位示意图

Slot	User may give as a constraint?
area	Yes, 15 possible values
children allowed	Yes, 2 possible values
food	Yes, 28 possible values
has internet	Yes, 2 possible values
has tv	Yes, 2 possible values
name	Yes, 163 possible values
near	Yes, 52 possible values
pricerange	Yes, 4 possible values
type	Yes, 3 possible values (restaurant, pub, coffeeshop)
addr	No
phone	No
postcode	No
price	No

图 6-7 DSTC3 中的对话槽位示意图

6.2.2 对话系统评测数据集

DSTC 系列数据集包括：

DSTC1[①]，2013 年的主题为 Bus Schedule

- 数据集为 3 年间匹斯堡公车路线电话自动查询系统的真实用户日志。

DSTC2，2014 年的主题为 Restaurant

- 目标可变：用户的目标在对话过程中可以改变。

DSTC3，2014 年的主题为 Restaurant + Tourist

- 迁移：bars 和 cafes 的训练数据很少，需要由 DSTC2 进行迁移学习。

DSTC4，2015 年的主题为 Tourist

- 主要针对旅游场景下人和人对话的内容进行评测。

DSTC5，2016 年的主题为 Tourist

① The DSTC1 data remains available at http://research.microsoft.com/en-us/events/ dstc/

- 主要针对跨语言对话建模的挑战。

DSTC6，2017 年的主题为端到端对话学习，建模及对话终止检测

- 任务型对话学习，人类对话模仿和对话终止检测。

DSTC7，2018 年的主题为答句选择、答句生成和视听场景感知对话

- 端到端的答句选择和答句生成，挑战结合实际场景的图片特征进行对话任务。

6.2.3 对话系统评测标准

Dialog DSTC 的评测指标如下。

假设准确率(Hypothesis Accuracy)：置信状态中首位假设(Top Hypothesis) 的准确率。此标准用以衡量首位假设的质量。

平均排序倒数：1/R 的平均值，其中 R 是第一条正确假设在置信状态中的排序。此标准用以衡量置信状态中排序的质量。

L2 范数（L2-norm）：置信状态的概率向量和真实状态的 0/1 向量之间的 L2 距离。此标准用以衡量置信状态中概率值的质量。

平均概率（Average Probability）：真实状态在置信状态中的概率得分的平均值。此标准用以衡量置信状态对真实状态的概率估计的质量。

ROC 表现（ROC Performance）：采取了两种 ROC 计算方式。在用第一种方式计算正确接受率(Correct Accept，CA)的比例时，分母是所有状态的总数。这种方式综合考虑了准确率和可区分度。在用第二种方式计算 CA 的比例时，分母是所有正确分类的状态数。这种计算方式单纯考虑可区分度而排除准确率的因素。

等误差率（Equal Error Rate）：错误接受率（False Accept，FA）和错误拒绝率（False Reject，FR）的相交点（FA=FR）。

正确接受率 5/10/20：当至多有 5%/10%/20%的 FA 时的 CA。

一个面向任务的对话系统的整体测评指标主要包括任务完成率和平均对话轮数两项，而对于各子模块，评测指标分别为：

NLU 模块的评测指标主要包括：分类问题、准确率、召回率和 F-score。

DST 模块的评测指标主要包括：参考关于 DSTC 的介绍。

DPL 模块的评测指标主要包括：任务完成率、平均对话轮数。

NLG 模块目前的主流实现技术为基于模板的方法，因此暂时不做评测。

6.3 闲聊系统评测

6.3.1 闲聊系统评测介绍

能否通过图灵测试可以被看作最早对闲聊系统进行评测的标准。但是，闲聊系统既不像问答系统那样有准确可查的参考答案，也不像面向任务的对话系统那样有明确的目的，因此，对闲聊系统的评测存在主观性、机变性等问题。目前尚无统一的对闲聊系统进行评测的数据集及评测标准。

张伟男等人[2]在 2017 年的一篇综述中给出了针对闲聊系统的评测方法，具体内容如下。

对闲聊系统进行评价的方法主要有客观指标评价与模拟人工评分两种。客观指标评价又可分为基于词重叠和词向量两种评价矩阵方法，BLEU[3]、METEOR[4]

和 ROUGE[5]是基于词重叠评价矩阵的代表；贪婪匹配法（Greedy Matching）[6]、向量均值法（Embedding Average）[7]、向量极值法（Vector Extrema）[8]则是典型的基于词向量的评价矩阵方法。模拟人工评分的思路采用神经网络模拟人工评分的方法，其中较有代表性的是谷歌的 Anjuli Kanan、Oriol Vinyals 等人提出的类 GAN 结构的对抗评价模型[9]、McGill 大学的 Ryan 等人[10]提出的基于 RNN 的自动对话评测模型（Automatic Dialogue Evaluation Model，ADEM）和基于人工神经网络（Artificial Neural Network，ANN）模型结构的对话评价系统。

6.3.2 闲聊系统评测标准

本节将详细介绍客观指标评测中常用的评测指标。基于词重叠的评测方法是自然语言处理任务中常用的评测方法，它通过计算系统生成的回复与标准答案中的词的重叠率来评测系统的效果。BLEU 和 METEOR 是机器翻译任务中应用广泛的两个基于词重叠的评测指标；ROUGE 是文本自动摘要任务中常用的评测标准。虽然上述指标不完全适用于闲聊系统的评测任务，但研究人员基于这些指标在闲聊系统的评测任务上进行了尝试和改进。

BLEU（Bilingual Evaluation Understudy）通过分析候选答案文本和参考答案文本中 N-gram 片段共同出现的次数来衡量系统的效果，于 2002 年由 IBM 提出。N-gram 表示 n 个连续单词的序列，即文本片段。BLEU 方法认为共现的文本片段数越多，模型的质量越好，并且这些文本片段与它们在上下文中的位置无关。BLEU 首先会对语料库中的所有语料进行 N-gram 的精确率计算：

$$p_n = \frac{\sum_{C \in \{\text{Candidates}\}} \sum_{\text{N-gram} \in C} \text{Count}_{\text{clip}}(\text{N-gram})}{\sum_{C' \in \{\text{Candidates}\}} \sum_{\text{N-gram}' \in C'} \text{Count}(\text{N-gram}')}$$

公式中，分子表示取 N-gram 在候选答案文本和参考答案文本中出现的最

小次数，分母表示取 N-gram 在候选答案文本中出现的次数。当候选答案文本很短时，N-gram 的精确率分值会很高，但实际上该答案可能并非一个很好的答案，因此针对候选答案文本比参考答案文本要短的情况，可以用一个惩罚因子 BP 去控制：

$$\mathrm{BP}=\begin{cases}1 & \text{若 } c>r\\ \mathrm{e}^{(1-r/c)} & \text{若 } c\leqslant r\end{cases}$$

其中 c 代表候选答案文本词数，r 代表参考答案文本词数，最终 BLEU 的公式为

$$\mathrm{BLEU}=\mathrm{BP}\cdot\exp\left(\sum_{n=1}^{N}w_n\log p_n\right)$$

W_n 是一个恒定的权重，N 表示 N-gram 中 n 的最大值。随着 N-gram 的增大，总体的精度得分呈指数下降，所以通常 N-gram 最多取到 4-gram，也就是 BLEU-4，这也是机器翻译任务中使用最广泛的指标。

METEOR 评测标准在 2004 年由 Lavir 等人提出。Lavir 等人通过研究发现，相比于单纯基于精确率的标准（如 BLEU），基于召回率的评测标准的评测结果和人工判断的结果有较高相关性。因此，METEOR 基于单精度的加权调和平均数和单字召回率综合计算，得出候选答案文本与参考答案文本之间的精确率和召回率的调和平均值 F-测度。METEOR 也包含一些其他指标没有的功能，如同义词匹配等，使用 WordNet 进行特定的序列匹配，如同义词、词根词缀、释义的校准。与 BLEU 不同，METEOR 同时考虑了基于整个语料库的精确率和召回率，才得出 F-测度。

ROUGE 是常用于自动生成文本摘要的一系列评价指标，包括 ROUGE-N、ROUGE-L、ROUGE-S、ROUGE-W、ROUGE-SU 等。举例来说，ROUGE-L 是通过统计候选答案文本与参考答案文本之间的最长公共子序列长度（Longest

Common Subsequence，LCS），再计算 F-测度得到的。LCS 是在两句话中都按相同次序出现的一组词序列。其与 BLEU 相似，都可以反映词语顺序，但是 ROUGE 的词可以是不连续的，而 BLEU 的 N-gram 要求词语必须连续出现。

除了词语的重叠率因素，另一种评价闲聊系统回复效果的思路是通过了解每一个词的语义来判断回复的准确性。词向量是实现这种评价方法的基础，通过采用前面介绍的 Word2vec 等方法，给每一个词分配一个向量用于表示该词，然后通过计算该词在语料库中出现的频率来近似地表示这个词所表达的含义。将一个句子中所有词的向量矩阵通过向量连接起来，就可以近似得到句子级的句向量。通过这种方法可以分别得到候选答案文本与参考答案文本的句向量，再通过各种距离计算方法得到两个句子的相似度。

贪婪匹配方法是一种基于词级别向量矩阵的匹配方法，对于给出的两个句子，先将每一个词都转化为词向量，然后将第一句中的每个词与第二句里的每个词做余弦相似度匹配，对全部结果进行加和求平均，得出的结果是所有词匹配之后的均值。

向量均值法是指通过句子中的每个词的词向量计算句子的向量表示的方法，即通过对句子中每个词的向量加和求均值获取句子的向量表示。候选回复和参考回复的相似程度可以通过计算两个句向量的余弦相似度进行评价。

向量极值法也是一种基于句向量计算候选回复和参考回复相似度的方法。该方法通过筛选句子中每个词的词向量组成的矩阵中每一维度的极值最大的值作为该维度的值，最终获得这个句子的向量表示。同样，还需要计算候选回复与参考回复之间的余弦距离才能表示它们之间的相似程度。某个文本中具有特殊意义的词应当比常用词拥有更高的优先级，但由于常用词往往会出现在更多的文本中，使得这些词在向量空间中距离更短，在计算相似度之后常用词会占据输出向量排序中靠前的位置，而这会使具有特殊意义的词被“挤”到靠后的

位置。因此，在采用向量极值法的时候，需要有意识地忽略常用词。

6.4 参考文献

[1] 吴友政, 赵军, 段湘煜, 等. 构建汉语问答系统评测平台.NCIRCS2004 第一届全国信息检索与内容安全学术会议论文集. 2004.

[2] 张伟男，张扬子，刘挺. 对话系统评价方法综述，中国科学: 信息科学，47（8）：953-966，2017.

[3] Papineni K, Roukos S, Ward T, et al. BLEU: A Method for Automatic Evaluation of Machine Translation.Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2002: 311-318.

[4] Banerjee S, Lavie A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments.Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization. 2005: 65-72.

[5] Lin C Y. Rouge: A Package for Automatic Evaluation of Summaries.Text Summarization Branches Out, 2004.

[6] Rus V, Lintean M. A Comparison of Greedy and Optimal Assessment of Natural Language Student Input Using Word-to-word Similarity Metrics. Proceedings of the Seventh Workshop on Building Educational Applications Using NLP. Association for Computational Linguistics, 2012: 157-162.

[7] Wieting J, Bansal M, Gimpel K, et al. Towards Universal Paraphrastic

Sentence Embeddings. arXiv preprint arXiv:1511.08198, 2015.

[8] Forgues G, Pineau J, Larchevêque J M, et al. Bootstrapping Dialog Systems with Word Embeddings. Nips, Modern Machine Learning and Natural Language Processing Workshop. 2014, 2.

[9] Kannan A, Vinyals O. Adversarial Evaluation of Dialogue Models. arXiv preprint arXiv:1701.08198, 2017.

[10] Lowe R, Noseworthy M, Serban I V, et al. Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses. arXiv preprint arXiv:1708.07149, 2017.

7 聊天机器人挑战与展望

7.1 开放式挑战

当前，聊天机器人的研究开展得如火如荼，但是其中仍然存在很多的挑战。例如，模块化系统的整合、端到端模型的建立和评价、对话策略的学习、聊天机器人评价等。

1. 模块化系统的整合

对于用户输入，传统的聊天机器人通过自动语音识别、自然语言理解、对话管理、答案检索、自然语言生成、语音合成等模块生成回复。不仅每一个具体模块都有其面临的问题与挑战，模块整合过程中模块间的兼容性及模块整合方式也是聊天机器人面临的挑战之一。例如，对用户输入语句完成自然语言理

解之后，如何在机器识别到错误的用户意图的情况下，保证对话管理模块可以修正错误的用户意图？用户情感分析、用户意图识别的结果应对生成的自然语言和合成的语音产生影响，避免错误传递，不能简单地通过将上述模块串行起来处理用户输入的问题。

2．端到端模型的建立和评价

端到端方法的目标是直接由用户输入得到输出，不经过传统方法中的自然语言理解、对话管理、自然语言生成等串行模块，因此端到端模型本身的好坏对对话质量起到决定性作用。首先，端到端模型需要充分考虑多轮对话中的对话管理和跟踪，回复的内容不仅需要考虑当前轮的用户输入，还需要考虑对话的上下文和语境；其次，端到端模型如何在生成回复时保持机器人个性的一致性也是一个难题，机器人个性一致会在很大程度上影响用户的体验，因此这是建立端到端模型时不容忽视的问题；另外，目前对端到端模型的评价并没有统一的标准，导致研究人员难以评定模型的好坏。研究人员一般基于某一公开的数据集训练自己的模型，且使用多种评测标准对比模型在该数据库上的表现。另一种评测方法是通过人工对话的形式，以用户的体验来判断模型质量，主观性很强。

3．对话策略的学习

目前，聊天机器人的对话策略一般为被动交互，也就是说等待用户输入唤醒机器。少数机器可以自动发起会话，但基本上是随机产生一些问题。对话策略学习最主要的一点就是让机器学习对话的主导方式。对话主导方式可以分为用户主导、机器主导和混合主导。机器学习对话主导方式的目标不仅是让机器在合适的时机主动主导对话，还包括优化机器主导对话时的话题选择和具体对话内容的选择。

4．聊天机器人评价

目前，对聊天机器人采用的基本是通用的客观评价标准，有回答正确率、任务完成率、对话轮数、对话时间、系统平均响应时间、错误信息率等，评价的基本单元都是单轮对话。但是，人机对话过程是一个连续的过程，而对不同聊天机器人系统的连续对话的评价仅能保证首句输入的一致性，当对话展开后，不同系统的回复不尽相同，因此不能简单地将连续对话切分成单轮对话去评价，于是设计合理的人工主观评价也许能够成为对聊天机器人系统智能程度评价的重要指标。

除了上述列举的具体问题，不论是基于传统方式还是基于端到端方式实现的聊天机器人，机器都需要保证机器人个性的一致、对上下轮对话状态的跟踪，以及尽量避免安全回答。

7.2　技术与应用展望

许多学者和评论家均认为未来聊天机器人会成为用户的私人助理，谷歌发布 api.ai，旨在解决语音识别、意图识别和语境管理的问题；Facebook 则致力于让聊天机器人“体会”用户的情绪。这些努力无疑将促进聊天机器人的发展和应用，但本轮聊天机器人的爆发能否带来人机交互方式的本质变革，尚无定论。归根结底，无论从技术还是从具体场景上分析，当前的聊天机器人技术仍然处于发展早期。

TechCrunch 的资深科技记者 Natasha Lomas 曾在 *TechCrunch* 上发文称，本轮聊天机器人的浪潮并不会带来所谓的范式改革，但是那些真正成功并且被社会保留下来的机器人，能做的绝不仅仅是聊天。结合 2016 年 Winograd Schema 挑战赛的结果（机器的最高准确率仅比随机概率高一点点），我们能猜测到聊天机器人发展过程中将面临非常巨大的挑战。纽约大学的研究心理学家、AI 初

创公司 Geometric Intelligence 的联合创始人 Gary Marcus 认为，人类需要与其他人进行有意义的情感互动，直到开发出真正的强人工智能，而机器的任务是帮人找到所需数据和互动对象。

需要注意的是，Facebook 研发聊天机器人并不是以让机器通过图灵测试为目标的。Facebook 旨在提供一种比安装其他应用或在网络上搜索更便捷的方式，帮助人们得到答案、满足用户需求。对大部分企业来说，一个聊天机器人只要能自动处理 30%的客户需求，便可以让企业愿意为聊天机器人的支出买单，因为机器已经能节约足够多的成本。因此，不论聊天机器人是否会成为用户的私人助理，也不论聊天机器人能否达到真正的人工智能，那些可以更便捷地满足用户需求或者为企业节省成本的聊天机器人都会被保留下来。

无论如何，随着聊天机器人研究的广泛开展，实现强人工智能成为广大学者和从业者的圣杯与目标，而为了实现强人工智能，未来对聊天机器人的研究将着眼于以下 4 个方面。

（1）从特定域到开放域：随着大数据和云时代的到来，开放域的聊天机器人系统更容易获取丰富的对话数据用于训练。

（2）更加关注“情商”：未来的聊天机器人研究将更注重“情商”，即聊天机器人对用户的个性化情感陪伴、心理疏导和精神安慰等能力。

（3）端到端对话系统：得益于深度学习技术的发展，端到端对话系统得到了广泛研究和应用，研究人员希望利用统一的模型代替序列化执行自然语言理解、对话管理和自然语言生成的步骤，从而根据用户的原始输入直接生成系统回复。

（4）更加关注现实：除了技术，还应关注伦理道德等更多方面。一个具体的例子就是微软的聊天机器人 Tay 由于在 Twitter 上发表脏话而被迫下线。导致 Tay 上线不到 24 小时就被迫下线的原因，并不仅是技术，更是现实情况的复杂性。

在商业落地方面，聊天机器人未来的实践可能会集中在以下几个方面。

1．客服机器人

特定领域乃至特定公司的服务场景是具体且固定的，其中用户会提出的问题、需要帮助的事情也是基本统一的。同时，企业往往拥有针对这些问题的高质量的问答对。因此，特定行业的客服机器人将在商业化实践方面有较成熟的应用，如各大银行的银行客服机器人、服务于政府的政务聊天机器人，以及服务于网络零售业务的店铺客服机器人等。

2．医疗机器人

医疗是一个具有大量规范数据和开放性研究的行业。机器通过“阅读”行业相关的研究文献和医疗记录，可以获得大量有用、可靠的数据和知识，基于这些数据和知识所进行的人机问答和对话将对医疗过程起到帮助，免去医生和专家检索文献和资料的时间。机器人在医疗行业最著名的应用就是本书前面章节介绍的 IBM Watson。

3．教育机器人

在教育这一具体场景中，有大量的资料可以作为资料库，供机器从中获得教育相关的基础知识、题库等。借助聊天这一具体交互方式，可以实现更好的教育效果。

各大厂商都推出了各自的教育聊天机器人，包括狗尾草智能科技的公子小白、科大讯飞的阿尔法蛋等。

随着自然语言处理、机器学习包括深度学习、知识图谱等技术的深度发展，聊天机器人会在几年内逐步走向成熟，并能够服务于人类生活的方方面面。但我们也看到聊天机器人存在着明显的短板，因此，未来聊天机器人将会如何发展，会不会有新的形态和品类出现，我们拭目以待。

7.3 从聊天机器人到虚拟生命

纵观聊天机器人发展史，最早的聊天机器人 ELIZA 被看作遵循符号主义的专家系统，当时的 AI 等价于逻辑（logic），所有的机器人都在一个特定领域通过一定逻辑的符号演算演绎来完成其功能。ELIZA 之后的聊天机器人 ALICE，被看作基于语言标记的聊天机器人，ALICE 的设计提示研究人员可以通过配置将聊天机器人的基本功能做好。随着配置的内容越来越多，聊天机器人的表现也越来越像一个真实的人。早期（10 年之前）的很多机器人，或多或少借鉴了 ELIZA 和 ALICE 的设计思想。2011 年，伴随着 iPhone 4S 出现的 Siri 的定位是个人助理，Siri 提高了虚拟个人助理的市场成熟度，使用户知道了虚拟个人助理，并且使用户逐渐习惯用语言与其进行交互。2011 年出现的 Watson 参加了《危险边缘》并打败了人类冠军。2012 年，受 Siri 的启发，Google 在智能移动终端的 Android 系统中推出了类似 Siri 的 Google Now。后来亚马逊又推出了以 Alexa 为技术倚靠的 Echo，再后来又有了 Cortana、ALLO、Tay 等。在聊天机器人发展的历史长河中，初期一般间隔很多年才会有更新换代的产品，但现在每年都会有很多相关的产品出现，各巨头都参与了关于聊天机器人的军备竞争。聊天机器人产品迭代的时间线如图 7-1 所示。

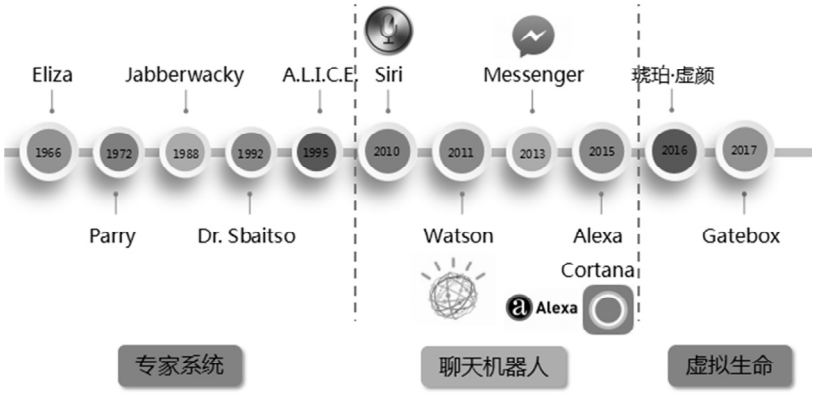

图 7-1 聊天机器人产品迭代的时间线

图 7-2 最右侧所示的虚拟生命是聊天机器人的发展趋势。从本质上理解，

虚拟生命是生命的延伸，具备生命的主要特征，包括感知能力、认知能力、自我进化的能力等。从感知能力的角度看，虚拟生命需要能听得到、看得见、可交互。从认知能力来看，虚拟生命能够和用户及周围环境进行“真实”“自然”的交流，包括具有规划、推理、联想、情感和学习的能力及可用性和可交互性。在达尔文的进化论中，物种是可变的、不断进化的，生物的变异、遗传和自然选择可以导致生物适应性的改变。

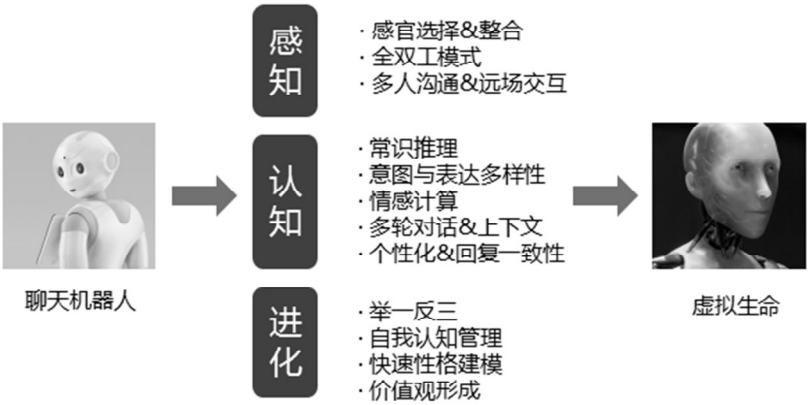

图 7-2　聊天机器人进化到虚拟生命所需要的能力

从技术的角度分析，上述的一切能力都依赖于语音识别、计算机视觉、语音合成、人工智能等技术的发展，从而使虚拟生命具有拟人性。以下我们给出一些里程碑性质的技术突破：2017 年 8 月 20 日，微软语音和对话研究团队负责人黄学东宣布微软语音识别系统取得重大突破，错误率由之前的 5.9%降低到 5.1%，可与专业速记员比肩[1]；Google 在 2015 年提出的深度学习算法，已经在 ImageNet 2012 分类数据集中将错误率降低到 4.94%，首次超越了人眼识别的错误率（约 5.1%）[2]；DeepMind 公司在 2017 年 6 月发布了目前世界上文本到语音环节做得最好的生成模型——WaveNet 语音合成系统①；由斯坦福大学发起的 SQuAD（Stanford Question Answering Dataset），截至 2018 年 12 月，使

① https://deepmind.com/blog/wavenet-generative-model-raw-audio/

用 BERT 的系统暂列第一，其 F1 分值达到 86.096；学霸君研发的 Aidam 机器人在 2017 年高考中取得了数学 134 分的高分。

从数据科学的角度看，来自卡内基梅隆大学的 William W. Cohen 教授指出，虽然大部分的自然语言处理问题都可以通过数据和机器学习（尤其是深度学习）来处理，但仍然有很多问题（比如说基于逻辑的语义解析）无法通过数据和机器学习得到解决。因此，可扩展性（Scalability）、表示（Representation）及机器学习（Machine Learning）作为数据科学的三个层面，虽然在融合上有一定困难，但三者的融合一定是未来的趋势。

聊天机器人和虚拟生命的发展依赖于自然语言处理，而大量的自然语言处理任务可转换为有监督的分类或序列标注问题。目前，我们往往会为特定任务下标注数据的缺乏或不充足发愁，这一点在利用深度学习时尤为严重。基于知识图谱的数据增强（Data Augmentation）对解决标注数据不足的问题具有显著意义。具体的做法是，将知识图谱与文本语料库关联形成大量弱标注数据。这种做法在关系抽取或事件抽取等任务上应用广泛。例如，对于三元组<琥珀,喜欢吃,葡萄>，通过将这个三元组进行一定的泛化，将琥珀转换为 PERSON，即在网络上收集 PERSON 和葡萄共现的描述片段，这些描述片段可能代表人物喜欢吃葡萄的特定模式，也可能代表噪声。为了提升泛化数据的整体质量，需要研究如何通过聚类分析中的异常点检测或噪声建模等方式识别并剔除弱标注语料中的噪声。

在虚拟生命的发展过程中，深度学习可以帮助我们研发出更多更有智慧的人工智能模型，使得我们可以更好地预测特定输入对应的输出；而知识图谱可以被看作虚拟生命的知识库，让虚拟生命变得更有学识，提升虚拟生命进行思考、语言理解、推理和联想的能力。随着技术的发展和数据的不断积累，如图 7-3 所示的由聊天机器人向虚拟生命发展的技术时代即将到来。

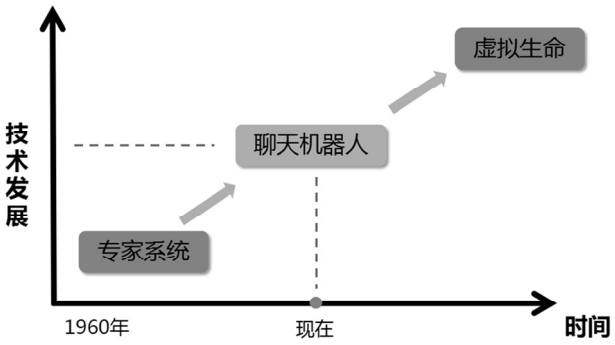

图 7-3　由聊天机器人向虚拟生命发展的技术时代

7.4　参考文献

[1] W. Xiong, L. Wu, F. Alleva, et al. The Microsoft 2017 Conversational Speech Recognition System, Microsoft Technical Report MSR-TR-2017-39, arXiv:1708.06073v2, 2017.

[2] K He, X Zhang, S Ren, et al. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,arXiv:1502.01852v1, 2015.